GRAPH CASCADES: Contagion-Based Mesoscopic Rewiring for Structure-Aware Graph Machine Learning

Meher Chaitanya[*]
KTH Royal Institute of Technology
Stockholm, Sweden
mcpi@kth.se

My Le[*]
Johns Hopkins University
Baltimore, MD
mle19@jh.edu

Luana Ruiz
Johns Hopkins University
Baltimore, MD
lrubini1@jh.edu

Abstract

We introduce GRAPH CASCADES, a mesoscopic rewiring strategy for Graph Neural Networks (GNNs) and Graph Transformers (GTs) that captures intermediate-scale graph structure beyond purely local edges or fully global attention. Using contagion-based diffusion with fixed cascade parameters, GRAPH CASCADES constructs, in $O(|V| + |E|)$ time, a sparse auxiliary graph where node pairs supported by repeated multi-hop reinforcement are promoted to direct neighbors. We theoretically characterize when reinforcement-based rewiring helps: sufficient conditions under which reinforcement-based edge selection is more label-aligned than direct adjacency, a stochastic block model (SBM) witness in which two-hop reinforcement is perfectly homophilic, and a formalization of mesoscopic connectivity via graph effective resistance. Empirically, across node-classification benchmarks, GRAPH CASCADES improves multiple GNN and sparse-GT backbones, with the most reliable gains observed on heterophilic and moderate- to high-degree homophilic graphs. The theoretical conditions also identify regimes where mesoscopic rewiring is unlikely to be beneficial—low-degree regular graphs and graphs with structural bottlenecks—and these predictions match the observed failures. We additionally observe tight correlations between performance and structural properties in the rewired graphs, suggesting that rewired graphs carry predictive signals.

1 Introduction

Graph-structured data arises across many domains, with applications including anomaly detection, traffic forecasting, and recommendation [Deng and Hooi, 2021, Kong et al., 2024, Agrawal et al., 2024]. A common strategy is to learn directly on the graph using Graph Neural Networks (GNNs), which capture local structure by propagating information along edges [Hamilton et al., 2017]. This locality is a strength, but repeated aggregation can blur signals and compress long paths, leading to oversmoothing and oversquashing [Chen et al., 2020, Alon and Yahav, 2021]. Graph Transformers (GTs), motivated by the success of transformers in language modeling, offer an alternative [Dwivedi and Bresson, 2020, Mialon et al., 2021]. GTs treat nodes as tokens under global self-attention, but since graphs lack a canonical ordering, attention alone cannot distinguish adjacent from multi-hop neighbors. GTs therefore concatenate positional or structural encodings (e.g., Laplacian eigenvectors, shortest-path distances) to each node, which has been central to their empirical gains [Kreuzer et al., 2021, Zhao et al., 2021, Ying et al., 2021].

Still, a key challenge is structural resolution: distinguishing and preserving task-relevant neighborhoods and motifs. Dense attention can overshadow these patterns, overweight weak node pairs, and

[*]Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

incur prohibitive cost [Ying et al., 2021, Chen et al., 2022, 2023, Zhou et al., 2025]. Sparse-attention GTs improve scalability by restricting attention to compact token sequences [Zhou et al., 2025], but existing variants sacrifice granularity through coarse neighborhood aggregation or hand-crafted shortcuts [Chen et al., 2023, Fu et al., 2024], an issue that is pronounced on heterophilic graphs, where same-label nodes are sparsely connected and multiple hops apart [Zhou et al., 2025]. To address these challenges, we introduce GRAPH CASCADES, a mesoscopic rewiring strategy that captures *intermediate-scale graph structure* via contagion dynamics [Granovetter, 1978, Centola and Macy, 2007, Centola, 2018]. Short threshold cascades are initiated at each node to reveal pairs that co-activate; these pairs are then promoted to direct connections in a sparse, weighted auxiliary graph that can be used either as a drop-in input for GNN and GT backbones or to instantiate CR-GRAPHORMER (Section 4), a lightweight standalone sparse transformer that directly acts on cascade-rewired neighborhoods.

GRAPH CASCADES is grounded in a concrete structural mechanism: it is expected to help when task-relevant relationships are expressed through reinforced multi-hop neighborhoods rather than primarily through direct edges. This setting arises naturally in heterophilic graphs, where nodes sharing a label need not be adjacent, yet may be connected through multiple short, redundant paths. In contrast, on low-degree regular graphs or graphs with pronounced bottlenecks, such redundant support may be weak or absent, making contagion-based rewiring neutral or even detrimental. Accordingly, our evaluation goes beyond predictive accuracy: we also examine the structural regimes in which the theory predicts gains or failures, treating this alignment between mechanism and performance as a central part of the contribution.

Theoretically, we provide a mechanistic account of why reinforcement-based rewiring can benefit heterophilic networks. We establish sufficient conditions under which reinforcement events exhibit higher label agreement than direct adjacency (Proposition 4.5); identify an SBM regime in which two-hop profile reinforcement recovers same-class neighbors (Theorem 4.7; and more generally multi-hop reinforcement in Appendix B.5) and characterize threshold activations through effective resistance (Theorem 4.8). These conditions also delineate regimes in which rewiring is unlikely to help, such as low-degree regular graphs and graphs with structural bottlenecks, and these predictions align with the observed failure cases. Empirically, cascade rewiring increases label homophily (Figure 2) and improves classification performance across heterophilic and moderate- to high-degree homophilic graphs (Tables 1–6, Figure 1), while structural properties of the rewired graph correlate strongly with downstream performance (Table 2).

2 Related Work

Graph Neural Networks and Transformers. Our main experiments evaluate GCN, GraphGPS, NAGphormer, and VCR-Graphormer with and without cascade rewiring. GraphGPS [Rampášek et al., 2022] interleaves message passing with global attention and replaces quadratic attention with linear mechanisms [Choromanski et al., 2021]. NAGphormer [Chen et al., 2023] constructs fixed-length hop-aggregate token lists and performs localized self-attention within each list, improving mini-batch efficiency and neighborhood encoding. VCR-Graphormer [Fu et al., 2024] explicitly targets heterophily by introducing virtual connections and PPR-based token lists that encode both local and global dependencies while maintaining linear complexity per layer. The broader comparison also includes GraphSAGE and GAT, together with GT [Dwivedi and Bresson, 2020], Gophormer [Zhao et al., 2021], Graphormer [Ying et al., 2021], SAN [Kreuzer et al., 2021], Specformer [Bo et al., 2023], Exphormer [Shirzad et al., 2023], SGFormer [Wu et al., 2023], and PolyFormer [Ma et al., 2024].

Diffusion and Graph Rewiring. PPRGo [Bojchevski et al., 2020] is a scalable GNN that decouples local feature transformation from graph propagation and then propagates predictions using precomputed sparse approximations of personalized PageRank (PPR), avoiding expensive multi-hop message passing or per-step power iteration while keeping long-range information. GRAND+ [Feng et al., 2022] is a scalable version of GRAND that precomputes a (generalized) mixed-order propagation matrix using a generalized forward push (GFPush) algorithm, so it can do GRAND-style random propagation graph augmentations in mini-batches, and it adds a confidence-aware consistency loss to improve generalization. Our mesoscopic rewiring conceptually falls under the same "rewiring" umbrella as diffusion-based methods (e.g., PPRGo/GRAND+), but it is explicit topology construction rather than an implicit propagation operator. GRAPH CASCADES is purely structural and label-free,

like FoSR [Karhadkar et al., 2023] and SDRF [Topping et al., 2022], but its objective is neighborhood realignment via reinforced multi-hop co-activation rather than bottleneck mitigation. Additional rewiring comparisons and an extended discussion of FoSR, SDRF, and GDC are provided in Appendix A.2.

3 Preliminary Definitions

Let $G = (V, E)$ be an undirected, unweighted graph with $n = |V|$ nodes and $m = |E|$ edges. For $u \in V$, denote $N(u) = \{v \in V : (u, v) \in E\}$, $N[u] = N(u) \cup \{u\}$, and $d(u) = |N(u)|$. A *contagion* or *cascade* on G is a discrete-time process $\{\mathbf{x}_t\}_{t=0}^{T}$, where $\mathbf{x}_t \in \{0, 1\}^n$ represents the instantaneous activation state at time t, with $[\mathbf{x}_t]_i = 1$ if node i is activated at time step t and 0 otherwise. The *active set* of a contagion at time t is defined as $S_t = \{u \in V : [\mathbf{x}_{t'}]_u = 1 \text{ for some } 0 \le t' \le t\}$, storing all nodes activated in the contagion by time t. Given an inactive node $u \notin S_t$, its instantaneous *active support* is the number of active nodes in its one-hop neighborhood, denoted $\sigma_t(u) = |N(u) \cap S_t|$.

In this paper, we leverage contagion dynamics to construct a mesoscopic rewiring by initiating a contagion with walk length ℓ from a neighborhood subset of each node $v \in V$. Each contagion evolves according to one of two discrete-time activation rules as defined in Chaitanya et al. [2025]:

Maximum Adjacency Search (MAS). For each node $v \in V$, select a subset $R \subseteq N(v)$ and initialize the cascade with active set $S_0^v = \{v\} \cup R$. At each step $t \in [0, \ell - 1]$, activate the inactive node with the strongest support $u_t^\star = \operatorname{argmax}_{u \in V \setminus S_t^v} \sigma_t^v(u)$ and set $S_{t+1}^v = S_t^v \cup \{u_t^\star\}$. The quantity $\tau_t = \max_{u \in V \setminus S_t^v} \sigma_t^v(u)$ is called the activation threshold at time t.

Threshold Adjacency Search (TAS). Fix a threshold $\tau \ge 1$. For each node $v \in V$, select a subset $R \subseteq N(v)$ and set $S_0^v = \{v\} \cup R$. At each step $t \in [0, \ell - 1]$, let $A_t = \left\{u \in V \setminus S_t^v : \sigma_t^v(u) \ge \tau\right\}$ be the set of inactive nodes with support at least τ. If $A_t \neq \varnothing$, select one $u_t^\star \in A_t$ (e.g., uniformly at random) and set $S_{t+1}^v = S_t^v \cup \{u_t^\star\}$. If $A_t = \varnothing$, the contagion process terminates. When $\tau = 1$ and activated nodes are processed in queue order, the process reduces to breadth-first search; larger τ restrict activation to nodes with multi-neighbor reinforcement [Granovetter, 1978, Centola, 2018].

Both MAS and TAS define higher-order activation processes. The next activation depends only on the current active set, and the walk length ℓ sets the spatial scale of propagation. MAS activation favors nodes with the strongest connectivity to the current active set, leading to early activation of densely connected neighborhoods, while nodes with weak ties are activated only as the threshold decreases. Thus, MAS tends to exhaustively explore cohesive regions before crossing weak cuts, causing the contagion to propagate preferentially within clusters. TAS relaxes this behavior by allowing nodes with lower instantaneous support to activate earlier, promoting a more balanced graph exploration that is less constrained by local density. Further preliminaries on GNNs, GTs, and sparse variants are provided in Appendix A.1.

4 Graph Cascades: Mesoscopic Rewiring

We construct the mesoscopic rewiring by leveraging the activation sets generated by the contagion dynamics in Section 3. For each seed $v \in V$, we run multiple contagions of fixed length ℓ, each initialized at v with a random subset $R \subseteq N(v)$. Each realization yields an activation set S_ℓ^v containing v, R, and up to ℓ nodes activated during the process. Aggregating activation sets across realizations defines a multiset $\mathcal{M}_v$ that records how frequently other nodes co-activate with v.

Let $f_v(u)$ denote the multiplicity of node u in $\mathcal{M}_v$ and retain the top-k co-activated nodes for each v:

$$F_k(v) = \underset{u \in V \setminus \{v\}}{\text{argtop-k}}\{f_v(u)\}. \tag{1}$$

If fewer than k nodes have positive co-activation counts $f_v(u) > 0$, then $F_k(v)$ contains only those positive-count nodes, so $|F_k(v)| \le k$. The corresponding activated edges are $E_{\text{act}}^v = \{(v, u) \mid u \in F_k(v)\}$. Repeating this for all $v \in V$, we define the undirected weighted *cascade-rewired auxiliary graph*:

$$G^* = (V, E^*, W^*), \qquad E^* = \bigcup_{v \in V} E_{\text{act}}^v, \qquad W_{uv}^* = \begin{cases} \frac{f_v(u) + f_u(v)}{2} & \text{if } (u, v) \in E^*, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The multiplicity $f_v(u)$ is the empirical cascade reinforcement score used for the directed top-k selection in (1); larger values indicate that u is repeatedly co-activated with seed v across cascade realizations. The undirected weight W^*_{uv} in (2) symmetrizes the two directed scores. In the theoretical analysis below, we use idealized short-walk reinforcement events as population-level proxies for this empirical co-activation score.

Remark 4.1 (Normalization)**.** *We may optionally normalize multiplicities before incorporating them into the weight matrix in* (2)*. This can be done either globally* ($f_v(u) \leftarrow f_v(u)/\max_{s,t\in V} f_s(t)$) *or locally* ($f_v(u) \leftarrow f_v(u)/\max_{t\in N(v)} f_v(t)$)*. The former preserves relative strength across the entire graph and emphasizes long-range structural effects, while the latter preserves relative importance within each node's neighborhood and ensures more uniform participation across nodes.*

By construction, the rewired graph G^* emphasizes cohesive communities and multi-path reinforcement, as both MAS and TAS contagion modes privilege multi-hop links in densely connected regions, where early activations correspond to nodes tightly coupled to the seed. Later activations identify peripheral or weakly connected nodes; consequently, nodes separated by weak cuts rarely appear in contagions with restricted walk length ℓ. This naturally reveals higher-order motifs and improves label homophily in heterophilic networks. We formalize this effect theoretically and validate it empirically in Section 4.1 and Appendix D.4.

The cascade-rewired auxiliary graph is the central contribution of this paper and can be used directly by GNN and GT backbones. To show that this structure also enables a standalone sparse GT, we introduce CR-GRAPHORMER, a lightweight model that constructs node-specific token sequences from cascade-rewired neighborhoods, making it suitable for settings in which the backbone cannot directly exploit edge weights. The model leverages the auxiliary graph G^* constructed from contagion dynamics to define compact, node-specific attention neighborhoods that capture higher-order structure beyond immediate adjacency.

CR-GRAPHORMER

Given an undirected, unweighted graph $G = (V, E)$ with node features $\mathbf{X} \in \mathbb{R}^{n\times d}$, we construct a *cascade-rewired* auxiliary graph $G^* = (V, E^*, W^*)$ following Section 4. To enable efficient mini-batch processing, we derive token sequences of length at most $k + 1$ from G^* for each target node v (see (1)), combining its original features $\mathbf{x}_v$ with those of its neighbors. These tokens are processed by a standard transformer [Vaswani et al., 2017] to produce a node representation vector. Formally, the token sequence is

$$\mathbf{T}_v := \Big(\underbrace{\mathbf{x}_v \| 1}_{\text{self}},\ \underbrace{\{W^*_{uv}\mathbf{x}_u \| W^*_{uv}\}_{u\in N^*(v)}}_{\text{neighbors in } G^*\text{, weighted}}\Big),$$

where $\mathbf{T}_v \in \mathbb{R}^{(k+1)\times(d+1)}$ and $\|$ denotes concatenation. Using $\mathbf{Z}_v^{(0)} = \mathbf{T}_v$ as the node-specific input sequence, each encoder layer applies standard transformer updates:

$$\tilde{\mathbf{Z}}_v^{(t)} = \mathrm{MHA}(\mathrm{LN}(\mathbf{Z}_v^{(t-1)})) + \mathbf{Z}_v^{(t-1)}, \quad \mathbf{Z}_v^{(t)} = \mathrm{FFN}(\mathrm{LN}(\tilde{\mathbf{Z}}_v^{(t)})) + \tilde{\mathbf{Z}}_v^{(t)},$$

where MHA is multi-head self-attention, FFN is a position-wise feed-forward network, and LN denotes layer normalization. After T layers, a readout (e.g., mean pooling) over $\mathbf{Z}_v^{(T)}$ produces the final node embedding. Since all tokens are pre-selected, training is mini-batchable with a sequence length of at most $k + 1$, independent of the original graph size, where k controls the node degrees in the auxiliary graph G^*.

4.1 Effect of Cascade Rewiring on Label Homophily

In heterophilic graphs, label information is often not carried by individual edges but by collections of multi-hop paths. Nodes sharing the same label may be non-adjacent, yet they can be repeatedly connected through higher-order multi-hop structures. This observation motivates replacing edge-level connectivity with cascade rewiring.

Setting and Theoretical Proxy. Let $G = (V, E)$ be an undirected graph with adjacency matrix A and labels $y : V \to \{1, \ldots, C\}$. The standard edge-level homophily is $h_G = \sum_{(u,v)\in E} \mathbf{1}\{y_u = y_v\}/|E|$.

Equivalently, for a uniformly random unordered pair of distinct nodes (u, v), let $S := \{y_u = y_v\}$ and $\mathcal{E} := \{A_{uv} = 1\}$. Then $h_G = \Pr[S \mid \mathcal{E}]$. The implemented rewiring selects, for each seed v, nodes with large empirical co-activation multiplicity $f_v(u)$. To analyze the structural mechanism behind this selection, we use the idealized short-walk reinforcement event $\mathcal{F} := \{(A^2 + \cdots + A^\ell)_{uv} \geq \kappa\}$, for fixed $\ell \geq 2$ and $\kappa \geq 1$. This event is not the exact finite-sample event defining G^*. Rather, it is a population-level proxy for the same reinforcement principle: node pairs supported by many short walks are more likely to receive high cascade co-activation scores. Define the corresponding reinforcement homophily as $h_{\mathcal{F}} := \Pr[S \mid \mathcal{F}]$. We next give a Bayes-rule sufficient condition under which this idealized reinforcement event is more homophilic than direct adjacency, i.e., $h_{\mathcal{F}} \geq h_G$.

Assumption 4.2 (Nondegeneracy)**.** $\Pr[\mathcal{S}] > 0$, $\Pr[\mathcal{E}] > 0$, *and* $\Pr[\mathcal{F}] > 0$. *This ensures that the label-agreement, edge-based, and reinforcement-based conditional probabilities are well-defined.*

Assumption 4.3 (Label-Reinforcement Alignment)**.** $\Pr[\mathcal{F} \mid \mathcal{S}] \geq \Pr[\mathcal{E} \mid \mathcal{S}]$. *That is, same-label node pairs are at least as likely to be short-walk reinforced as they are to be directly adjacent.*

Assumption 4.4 (Selectivity of Reinforcement)**.** $\Pr[\mathcal{F}] \leq \Pr[\mathcal{E}]$. *This rules out saturated regimes in which the reinforcement event occurs for most node pairs and therefore becomes less selective than direct adjacency.*

Proposition 4.5 (Bayes Sufficient Condition for Reinforcement Homophily)**.** *Under Assumptions 4.2 and 4.3, $h_{\mathcal{F}} \geq h_G \cdot \Pr[\mathcal{E}] / \Pr[\mathcal{F}]$. If Assumption 4.4 also holds, then $h_{\mathcal{F}} \geq h_G$. The inequality $h_{\mathcal{F}} \geq h_G$ is strict if Assumption 4.3 holds strictly, or if Assumption 4.4 holds strictly and $h_G > 0$.*

Proof. See Appendix B.1. □

Proposition 4.5 is algorithm-agnostic: the event $\mathcal{F}$ is an idealized reinforcement-based selection rule, and the analogous selection in the implemented method is induced by retaining high-$f_v(u)$ cascade neighbors. The proposition identifies a sufficient condition under which any such rule has at least the label agreement of the original adjacency, namely that it be sufficiently label-aligned (Assumption 4.3) and sufficiently selective (Assumption 4.4). Appendix D.4 verifies this empirically: cascade rewiring is most beneficial on heterophilic graphs, where label information is carried by reinforced multi-hop structure rather than direct edges, and less reliable on low-degree homophilic graphs, where sparse local structure provides limited same-label reinforcement.

Remark 4.6 (Finite-Sample Cascade Proxy)**.** *The rewired graph G^* is the finite-sample top-k graph induced by empirical scores $\widehat{p}_v(u) = f_v(u)/N_v$, where N_v is the number of cascade realizations used for seed v. Let $p_v(u) = \mathbb{E}[\widehat{p}_v(u) \mid G]$. Under the independent-block sampling assumptions of Theorem B.6, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, simultaneously for all seeds v with $N_v \geq 1$ and all $u \neq v$, $|\widehat{p}_v(u) - p_v(u)| \leq \varepsilon_v(\delta) := \sqrt{\frac{\log(2n(n-1)/\delta)}{2N_{\text{eff}}(v)}}$. Here $N_{\text{eff}}(v) = N_v^2 / \sum_j a_{v,j}^2$, where $a_{v,j}$ are the sizes of the independent blocks of realizations for seed v; dependence within each block is permitted. Consequently, the empirical and population top-k sets coincide whenever the gap between the k-th and $(k+1)$-th population scores is strictly greater than $2\varepsilon_v(\delta)$. Moreover, for fixed-threshold TAS, every nonlocal positive-count selection $v \to u$ certifies $(A^2 + \cdots + A^{\ell+1})_{vu} \geq \tau$ (see Theorem B.6 for a precise statement). If counts are aggregated over thresholds $\mathcal{T}$, the unconditional certificate uses $\tau_{\min} = \min \mathcal{T}$; threshold-specific counters certify their corresponding threshold. Thus, G^* should be viewed as a finite-sample cascade analogue of the idealized reinforcement graph induced by $\mathcal{F}$ rather than an exact realization of it. We measure homophily on the rewired graph via $h_{G^*} = \Pr[y_u = y_v \mid (u, v) \in E^*]$, and compare this estimate with the population proxy $h_{\mathcal{F}}$ in Appendix D.4.*

Case Study: Stochastic Block Model (SBM) as a Witness for the Reinforcement Mechanism

To make the reinforcement mechanism concrete, we analyze a minimal SBM setting in which direct adjacency and short-walk reinforcement can disagree. Direct edges are governed by the block-affinity matrix B, whereas expected two-hop reinforcement $\mathbb{E}[A^2_{uv}]$ is governed by the profile-similarity kernel $BD_\pi B^\top$. Thus, the SBM construction serves as a witness that a graph may be edge-heterophilic while its reinforced two-hop neighborhoods are homophilic. This abstraction removes finite sampling, initialization, and MAS/TAS-specific effects while preserving the core reinforcement principle used by GRAPH CASCADES: pairs supported by many short paths are promoted to direct neighbors.

Setting. Let $G \sim SBM(n, \pi, B)$ be a C-block stochastic block model with node set V, labels $y_v \in [C]$, block sizes $n_a = \pi_a n$, and symmetric block-affinity matrix $B \in [0,1]^{C\times C}$ where C, π, and B are fixed and are independent of n. Conditional on the labels, edges are independent and $\Pr[A_{uv} = 1 \mid y_u = a,\ y_v = b] = B_{ab}$. For two distinct nodes u, v, define the two-hop cascade reinforcement score $s_2(u, v) := (A^2)_{uv} = |N(u) \cap N(v)|$. For each node u, define the directed top-k two-hop cascade graph by connecting u to the k nodes $v \neq u$ with the largest $s_2(u, v)$, and let $G_2^\star(k)$ denote its symmetrization. For labels $a, b \in [C]$, define $\mu_{ab}^{(n)} := \mathbb{E}[s_2(u, v) \mid y_u = a, y_v = b]$ with $u \neq v$. The following theorem assumes that same-class pairs are uniformly separated from different-class pairs.

Theorem 4.7 (Profile Homophily of Two-Hop Top-k Selection)**.** *Under the SBM setting above with $D_\pi := \mathrm{diag}(\pi_1, \ldots, \pi_C)$, $\frac{1}{n}\mu_{ab}^{(n)} = (BD_\pi B^\top)_{ab} + O\big(\frac{1}{n}\big)$ for every $a, b \in [C]$. Assume the profile kernel $BD_\pi B^\top$ has a positive same-class margin $\Delta_2 := \min_{a\in[C]}\big((BD_\pi B^\top)_{aa} - \max_{b\neq a}(BD_\pi B^\top)_{ab}\big) > 0$. Then for all sufficiently large n, with probability at least $1 - 2n^{-2}$, $s_2(u, v) > s_2(u, w)$ for every seed node $u \in V$ and any other nodes $v, w \in V \setminus \{u\}$ with $y_v = y_u$ and $y_w \neq y_u$. Consequently, for every $k \leq \min_{a\in[C]}(n_a - 1)$, the symmetrized top-$k$ two-hop cascade graph $G_2^*(k)$ is perfectly homophilic: $h_{G_2^*(k)} = 1$.*

Proof. See Appendix B.2. □

Theorem 4.7 isolates the reinforcement mechanism rather than analyzing the finite-sample MAS/TAS pipeline. Even when B favors cross-label edges, the profile kernel $BD_\pi B^\top$ can separate same- from different-label pairs, so the top-k two-hop rewired graph is perfectly homophilic with high probability. The witness can still be edge-heterophilic before rewiring: $h_G \xrightarrow{p} \sum_{a=1}^C \pi_a^2 B_{aa} / \sum_{a,b=1}^C \pi_a \pi_b B_{ab}$, which falls below $1/2$ whenever cross-class edge mass dominates (Corollary B.2). Corollary B.3 shows that, in a contextual SBM with Gaussian class-conditional features, one-hop aggregation over the top-k two-hop rewired graph followed by a nearest-centroid linear classifier has expected misclassification at most $2n^{-2} + (C-1)\exp\Big(-\frac{(k+1)\Delta^2}{8\sigma^2}\Big)$; where Δ is the smallest distance between class means and σ^2 is the feature noise variance; the $k+1$ factor links the top-k parameter directly to the downstream signal-to-noise ratio. The multi-hop extension of Theorem 4.7 appears in Appendix B.5, Theorem B.5. Corollary B.8 provides an end-to-end guarantee for a one-round, parallel, nonlocal-only TAS variant. Under the stated SBM and sampling assumptions, its empirical top-k graph is perfectly homophilic with high probability and supports a Gaussian-feature classification bound using uniform averaging and the true class means. Extending this guarantee to the sequential multi-step MAS/TAS schedule remains open.

4.2 Choice of Walk Length ℓ

The walk length ℓ sets the spatial scale at which $f_v(u)$ accumulates support. We use effective resistance [Spielman and Srivastava, 2008] as a structural certificate for threshold activations: if a node is activated within ℓ steps by a threshold-κ cascade, then its activation is supported by multiple short paths from the initial active set. This is a one-way statement about activated nodes, not a guarantee that all low-resistance nodes are selected. It directly describes TAS-style thresholding; for MAS, an analogous bound holds whenever the activation threshold τ_t is at least κ.

Setting. Let $G = (V, E)$ be an undirected, unweighted graph. Fix a nonempty initial active set $S_0^v \subseteq V$, and let $S_0^v \subseteq S_1^v \subseteq \cdots \subseteq S_\ell^v$ be the active sets generated by a uniform threshold-κ contagion, where $\kappa \in \mathbb{N}$. That is, whenever a vertex x first activates at time $t \geq 1$, it satisfies $\big|N(x) \cap S_{t-1}^v\big| \geq \kappa$. Let $A^v := \bigcup_{t=0}^\ell S_t^v = S_\ell^v$. For any $u \in S_\ell^v \setminus S_0^v$, let $T(u) := \min\{t \in \{0, 1, \ldots, \ell\} : u \in S_t^v\}$ be its activation time. Let G_v be the subgraph induced by the activated nodes $\bigcup_{t=0}^\ell S_t^v$. We write $R_{\mathrm{eff}}^G(S, u)$ for the effective resistance between u and the terminal obtained by contracting all vertices of S to a single vertex.

Theorem 4.8 (Effective-Resistance Bound for Threshold Activations)**.** $R_{\mathrm{eff}}^{G_v}(S_0^v, u) \leq \frac{T(u)}{\kappa} \leq \frac{\ell}{\kappa}$.

Proof. The proof follows directly from the two lemmas below (see Appendix B.8).

Lemma 4.9. *Following the setting of Theorem 4.8, if $u \in S_\ell^v \setminus S_0^v$, then G_v contains κ pairwise edge-disjoint paths from S_0^v to u. Moreover, each of these paths has length at most $T(u)$, and hence at most ℓ.*

Proof. See Appendix B.6.

Lemma 4.10. *Let $H = (W, F)$ be an undirected, unweighted graph, let $S \subseteq W$ be nonempty, and let $u \in W \setminus S$. Suppose H contains κ pairwise edge-disjoint paths $P_1, \dots, P_\kappa$ of lengths $\ell_1, \dots, \ell_\kappa$ from S to u. Then $R_{\text{eff}}^H(S, u) \le \frac{1}{\kappa^2} \sum_{i=1}^{\kappa} \ell_i$. In particular, if each $\ell_i \le L$, then $R_{\text{eff}}^H(S, u) \le \frac{L}{\kappa}$.*

Proof. See Appendix B.7.

Theorem 4.8 implies that activated nodes are not arbitrary long-range shortcuts: they admit κ edge-disjoint short paths from the initial active set, and the resulting R_{eff} bound scales with ℓ. While this is a one-way certificate rather than a global optimization guarantee, it motivates using a small ℓ in our experiments to concentrate $f_v(u)$ on mesoscopically well-supported neighbors.

Edge-Connectivity Certificate. A complementary certificate follows from the same activation-path argument. Appendix B.10 shows that any nonlocal directed edge $v \to u$ selected by a fixed-threshold TAS-κ run satisfies $\lambda_G(N[v], u) \ge \kappa$ in the original graph, where $\lambda_G(N[v], u)$ denotes the set-to-vertex edge connectivity (the maximum number of edge-disjoint paths from $N[v]$ to u). Intuitively, fixed-threshold TAS rewiring reinforces redundant mesoscopic connectivity and cannot introduce propagation-selected shortcuts across cuts of size smaller than κ. Together, these results delineate the regime where cascade rewiring is expected to help—graphs whose label-relevant structure is carried by reinforced multi-hop neighborhoods with sufficient local redundancy—and identify bottlenecked sparse graphs as the predicted failure mode.

5 Experiments

In this section, we show how our cascade-based mesoscopic rewiring improves homophily in heterophilic graphs and boosts the performance of GNNs and GTs. Additionally, we show that our proposed CR-GRAPHORMER is a competitive alternative to existing graph models.

Datasets. We consider 16 benchmarks from the Deep Graph Library (DGL)[2] including both homophilic and heterophilic settings. Our code is publicly available.[3] Dataset information is given in Table 7 (Appendix D.1). To study the effects of mesoscopic cascades, we generate 125 balanced-community SBM networks with a fixed intra-community edge probability $p_{\text{in}} = 0.005$. We vary the number of nodes $n \in \{500, 1000, 1500, 2000, 2500\}$, the number of communities $c \in \{2, 4, 6, 8, 10\}$, and the inter-community edge probability $p_{\text{out}} \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ to cover a range of network structures. Node features are sampled i.i.d. as $\mathbf{X} \sim \mathcal{N}(0, 1)^{n \times 5}$. We use these synthesized graphs to examine how mesoscopic cascades affect label homophily (Appendix D.4) and assess the correlation between model performance and graph structural statistics (Table 2). Additional experiments on long-range graph benchmarks (*PascalVOC-SP*, *COCO-SP*, *Questions*, *Roman-empire*, and *Amazon-ratings*) and non-duplicate heterophilic datasets (*chameleon-filtered* and *squirrel-filtered*) are reported in Appendix D.9 with the corresponding standard metrics (accuracy, macro-F1, and AUROC).

Mesoscopic Cascades. We rewire graphs using the MAS and TAS cascades from Section 3, with initial active-set size $|R| = 5$ and cascade length $\ell = 10$, each with $P = 5$ permutations and k in (1) set to the rounded average degree to preserve sparsity. For TAS, we use thresholds $\tau \in \{1, 2, 3, 4, 5\}$ and select the frequency normalization by validation performance (Remark 4.1). *This is our default setting.* Methods with MAS/TAS in their names use the corresponding weighted rewired graph (G^*, W^*); all others use the original unweighted graph G. From Appendix D.6, TAS is generally preferable on heterophilic graphs because thresholding filters weakly supported activations, while MAS gives tighter structural alignment but is more sensitive to walk length. On skewed-degree graphs, setting k to the median degree rather than the average may reduce over-inclusion.

[2] `https://www.dgl.ai/dgl_docs/api/python/dgl.data.html`

[3] GitHub: `https://github.com/ruiz-lab/Graph-Cascades`

Experimental Details. We compare against sixteen representative methods, comprising five GNNs and eleven GTs, with details provided in Appendix D.2. Each model is trained for up to 2000 epochs. All experiments use a 50%–25%–25% train–validation–test split and are repeated over 20 random splits per graph for robustness. Table 4 reports the within-backbone comparison for GCN, GraphGPS (GPS), NAGphormer (NAG), and VCR-Graphormer (VCR) along with our CR-GRAPHORMER. Table 6 provides the broader cross-architecture context in comparison with our best validation selected cascade models (the configuration with the highest mean validation accuracy). We report Tables 4 and 6 without standard deviations for readability; complete numerical results with standard deviations are in Appendix D.3 (Tables 8 and 9). Complete experimental details, the empirical validation of Assumptions 4.2–4.4 and Proposition 4.5, ablation studies, and comparisons with other rewiring methods and heterophily-specific architectures are provided in Appendices D.1, D.4, D.6, D.7 and D.8, respectively.

Table 1: Number of DGL graphs where GRAPH CASCADES improves test accuracy (GraphGPS counts exclude 3 OOM target datasets). Colors denote the graph type with the largest gain: **red** (heterophilic, target regime), **blue** (moderate- to high-degree homophilic, target regime), and **black** (low-degree homophilic). Complete results are in Table 4.

Baseline	Cascade Model	Largest Increase	# Graphs Improved Out of All 16 Graphs	# Graphs Improved Out of 12 Target Graphs
GCN	GCN-MAS	**+17.11%**	8 (avg. +0.39%)	8 (avg. +1.06%)
	GCN-TAS	**+20.15%**	10 (avg. +2.87%)	10 (avg. +4.03%)
GraphGPS	GPS-MAS	**+8.15%**	8 (avg. +2.00%)	7 (avg. +3.29%)
	GPS-TAS	**+9.52%**	9 (avg. +2.23%)	7 (avg. +3.08%)
NAG	NAG-MAS	**+19.77%**	11 (avg. +4.67%)	11 (avg. +8.85%)
	NAG-TAS	**+34.54%**	13 (avg. +6.94%)	12 (avg. +11.66%)
VCR	VCR-MAS	**+32.74%**	13 (avg. +10.41%)	12 (avg. +15.90%)
	VCR-TAS	**+32.33%**	13 (avg. +13.63%)	12 (avg. +19.79%)

Table 2: Regression coefficients for $A(M) \sim H * L(D) + H * L(D) * C$, where H is label homophily, $L(D)$ is log average degree, and $C \in \{0, 1\}$ indicates original graph connectivity. $A(M)$ denotes test accuracy, evaluated on the original graph unless MAS/TAS specifies an auxiliary one.

M	β_0	β_1	β_2	Adjusted R^2
NAG	0.193***	0.197***	0.170**	0.454
NAG-MAS	0.143***	0.258***	0.173***	0.722
NAG-TAS	0.152***	0.265***	0.237***	0.695
VCR	0.253***	0.233***	-0.005	0.426
VCR-MAS	0.155***	0.306***	0.157***	0.704
VCR-TAS	0.172***	0.301***	0.201***	0.607
CR-MAS	0.091***	0.238***	0.261***	0.760
CR-TAS	0.132***	0.219***	0.421***	0.639

$^{***}p < 0.001$; $^{**}p < 0.01$; none: $p \geq 0.1$

Table 3: Alphabetical dataset labels and names. Colors indicate graph type, aligned with Table 1. See Table 7 for exact label homophily and average degree.

A	*actor*	**B**	*chameleon*	**C**	*citeseer*	**D**	*community*	**E**	*computer*	**F**	*cora*	**G**	*cornell*	**H**	*cycle*
I	*grid*	**J**	*photo*	**K**	*pubmed*	**L**	*shape*	**M**	*squirrel*	**N**	*texas*	**P**	*wiki*	**Q**	*wisconsin*

Table 4: Test accuracy (mean, in %) across DGL datasets, summarized in Table 1. OOM indicates out-of-memory errors. Accuracy is encoded by shading, with darker cells indicating better performance. Table 3 provides dataset names. Table 9 reports the corresponding standard deviations.

	Model	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q
GNN-based Architectures	GCN	29.59	57.30	75.70	55.63	90.30	87.96	45.11	60.73	59.34	93.90	87.16	89.46	39.10	54.57	83.89	51.41
	GCN-MAS	34.28	57.67	74.71	51.76	89.34	84.93	60.85	59.27	58.30	93.91	87.87	57.69	40.01	66.49	81.79	68.52
	GCN-TAS	34.62	58.75	75.46	50.11	90.87	87.29	57.02	60.39	58.19	94.17	88.15	89.57	42.38	65.00	83.54	71.56
	GraphGPS	35.19	60.78	75.23	44.00	OOM	86.65	61.38	55.11	57.41	94.64	OOM	46.74	40.04	69.68	OOM	72.66
	GPS-MAS	36.39	59.90	74.09	47.43	OOM	85.49	68.72	55.78	55.44	94.71	OOM	54.89	39.20	75.00	OOM	78.44
	GPS-TAS	36.46	59.66	75.28	45.26	OOM	86.53	67.23	57.51	56.33	94.84	OOM	56.26	39.18	75.64	OOM	78.28
Tokenized Sparse-Attention GTs	NAG	31.68	40.38	74.72	46.17	88.66	86.75	58.94	77.24	70.99	93.94	87.26	49.89	25.80	63.30	82.40	61.25
	NAG-MAS	35.10	59.15	72.17	52.59	89.10	84.84	74.36	59.00	62.25	94.27	88.19	57.54	40.89	81.28	82.38	81.02
	NAG-TAS	35.10	58.41	74.15	58.21	89.19	86.82	74.36	59.20	60.60	94.61	88.26	84.43	42.13	81.28	82.59	81.02
	VCR	26.41	27.28	71.15	28.13	82.45	85.41	44.89	61.64	58.32	88.45	83.25	44.66	22.48	54.68	78.46	51.33
	VCR-MAS	36.18	60.02	71.47	47.22	89.43	85.08	73.85	51.70	44.05	93.88	87.59	52.12	39.37	80.99	82.68	79.93
	VCR-TAS	36.28	59.61	74.14	57.97	88.95	85.11	75.02	50.09	47.84	94.22	88.70	76.94	46.06	82.31	82.84	81.00
	CR-MAS	31.52	62.82	50.63	54.04	89.44	79.81	54.04	59.18	61.80	93.73	87.53	54.37	45.72	65.43	82.65	66.56
	CR-TAS	34.64	63.65	51.84	78.60	89.33	81.02	55.74	61.32	60.74	93.95	87.75	85.94	47.31	65.43	83.04	70.39

Model-Agnostic Gains from Cascades. From Tables 1 and 4, cascade rewiring with just default parameter selections already improves classification on a substantial portion of benchmarks for both GNNs and GTs. The gains are most pronounced and frequent for NAG and VCR, where cascades improve performance on up to $13/16$ graphs, indicating that these architectures benefit particularly strongly from cascade rewiring. Notably, across the 12 graphs in the target regimes, MAS improves NAG and VCR performance by an average of **+8.85%** and **+15.90%**, while TAS increases performance by **+11.66%** and **+19.79%**, respectively. Improvements are also widespread for GCN ($10/12$ target graphs) and consistent for GraphGPS ($7/9$ target graphs; the remaining 3 are OOM). Overall, the strongest single improvements reach **+32.74%** under MAS and **+34.54%** under TAS.

We observe pronounced improvements on all 6 heterophilic datasets (*actor*: +9.87%; *chameleon*: +32.74%; *cornell*: +30.13%; *squirrel*: +23.58%; *texas*: +27.63%; *wisconsin*: +29.67%). A possible

explanation for this—and for the performance declines in homophilic networks such as *citeseer*, *cora*, *cycle*, and *grid*—is that cascades primarily boost homophily in heterophilic and moderate- to high-degree homophilic graphs but decrease homophily in low-degree homophilic graphs due to the violation of Assumption 4.3. This trend aligns with Proposition 4.5 and is consistent across GNNs and GTs (Figure 1 in Appendix D.3, Table 10 in Appendix D.4). Table 5 summarizes the regimes induced by Assumptions 4.3 and 4.4 and the empirical behavior they predict; *the assumptions are evaluated directly on each DGL benchmark* in Appendix D.4.

Table 5: Regimes identified by alignment (Assumption 4.3) and selectivity (Assumption 4.4) conditions of Proposition 4.5, together with their empirical behavior. Full per-dataset verification is provided in Appendix D.4.

Regime	**Alignment**	**Selectivity**	**Example Datasets**	**Observed behavior**
Heterophilic	Holds	Holds	*actor*, *chameleon*, *cornell*, *squirrel*, *texas*, *wisconsin*, *Amazon-ratings*, *Questions*	Largest gains ($+9.87\%$ to $+32.74\%$)
Moderate to high-degree homophilic	Holds	Fails	*community*, *computer*, *photo*, *pubmed*, *shape*, *wiki*, *COCO-SP*, *PascalVOC-SP*	Often positive gains via Theorem 4.8
Low-degree homophilic	Fails	Holds	*citeseer*, *cora*	Small or unstable changes
Bottlenecked or highly regular	Fails	N/A	*cycle*, *grid*, *Roman-empire*	Predicted failure mode, unstable changes

Furthermore, cascade rewiring improves macro-F1 on the cohesive mesoscopic long-range datasets *COCO-SP* and *PascalVOC-SP*, improves AUROC on *Amazon-ratings* and *Questions*, and yields gains on *chameleon-filtered*. On *Roman-empire*, the effect is backbone-dependent: GNN-based architectures degrade, consistent with the scarcity of the κ edge-disjoint short paths required by Theorem 4.8 and Appendix B.10, whereas sparse-attention GT backbones gain, modestly for NAG (up to $+4.09\%$) and strongly for VCR (up to $+39.05\%$). See Appendix D.9 for numerical results. Results across six additional backbones (GAT, GT, Gophormer, ACMGCN, H2GCN, and GloGNN) on nine graphs in the target regimes (heterophilic and moderate- to high-degree homophilic) are reported in Table 25 (Appendix D.10) and are consistent with the trends in Tables 1 and 4.

Remark 5.1 (A Note on Cascade Weights)**.** *The weights W^* are intrinsic to cascade rewiring: they are the empirical reinforcement scores produced by the contagion dynamics, not a post-hoc addition. Thus,* GRAPH CASCADES *is best viewed as a single weighted operator. The ablations in Appendix D.6.2 (Tables 12 and 13) show that the role of W^* depends on the backbone. For sparse-attention GTs, the unweighted graph G^* already gives broad gains, indicating that the rewired topology carries meaningful structural signal. For GNN backbones, whose aggregation is more sensitive to degree imbalance, the frequency weights provide larger and more consistent improvements. We therefore use the weighted operator as the default configuration. The topology-only ablation confirms that both structure and reinforcement strength contribute to performance.*

How Accuracy Relates to Homophily, Degree, and Connectivity. To further assess the impact of cascade rewiring on graph learning performance, we conduct a regression study relating test accuracy to key graph characteristics, including label homophily, average degree, and graph connectivity. These statistics are computed on the input graph: the original graph with unit edge weights for baseline methods, and the MAS/TAS-rewired graphs with frequency-based weights.

Table 2 reports regressions of test accuracy on topological metrics across 125 synthesized graphs and 16 benchmark datasets. Overall, rewired structural properties explain a larger fraction of performance variation than those derived from the original graphs. While TAS typically yields larger absolute performance gains across datasets and architectures (Tables 1 and 4), MAS shows stronger alignment with the topological structure, attaining higher adjusted R^2 values for NAG, VCR, and the highest overall for CR-GRAPHORMER. Notably, despite synthesized graphs using random node features with no semantic content, we observe consistent correlations between accuracy and structural metrics across both synthesized and real-world benchmarks (Figure 3 in Appendix D.5). This indicates that the observed effects are primarily driven by graph topology rather than feature semantics.

Comparison to GNN and GT Baselines. We compare our best validation selected cascade models against the five GNNs and eleven GTs. The best validation-selected cascade model ranks among the top five on all 16 datasets and first on six (Table 6). While the test accuracy of our best model is lower

than the highest accuracy from baseline GNNs and GTs by 20.15% on *cycle* and 9.89% on *grid*, it remains competitive on the other graphs (***at most 1.18% lower than the observed best accuracy of all GNN and GT baselines***). The declines on *cycle* and *grid* may reflect their low degree and regular structure, which limit the benefits of cascade rewiring.

Table 6: Test accuracy (mean, %) on DGL datasets. OOM denotes out-of-memory. "Best-Validated" indicates the best validation-selected cascade models from Table 4. Top-5 models are shaded (rows 1-17 only, darker indicates better performance). See Table 3 for dataset names and Tables 8 and 9 for standard deviations.

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q
GNNs	GCN	29.59	57.30	75.70	55.63	90.30	87.96	45.11	60.73	59.34	93.90	87.16	89.46	39.10	54.57	83.89	51.41
	GraphSAGE	34.57	64.36	75.88	52.09	90.39	88.09	64.47	59.27	58.19	95.31	87.33	43.11	45.41	79.79	84.57	72.89
	GAT	28.85	58.63	75.77	58.99	OOM	87.38	44.15	59.27	58.19	94.49	86.88	43.11	OOM	56.70	OOM	49.45
	PPRGo	31.36	45.83	74.33	39.43	91.13	84.30	46.91	59.27	58.19	95.09	86.72	43.11	30.22	58.51	84.16	55.63
	GRAND+	30.20	45.54	73.80	39.37	90.85	83.70	45.11	59.27	58.19	94.82	85.95	43.11	30.23	56.49	83.38	53.20
GTs	GT	34.19	64.83	68.60	54.00	OOM	85.41	61.91	56.32	54.64	OOM	86.59	23.23	OOM	77.77	OOM	73.52
	Gophormer	34.90	53.16	31.48	71.00	90.60	49.18	67.13	54.98	62.62	94.98	40.40	79.83	37.44	81.49	83.34	81.95
	Graphormer	35.01	50.70	33.18	73.20	OOM	35.89	51.49	81.48	72.14	89.02	OOM	85.57	40.55	68.62	OOM	71.33
	SAN	36.04	59.22	73.37	46.00	OOM	85.07	67.55	59.27	57.27	94.85	OOM	29.40	39.00	80.00	OOM	80.63
	GraphGPS	35.19	60.78	75.23	44.00	OOM	86.65	61.38	55.11	57.41	94.64	OOM	46.74	40.04	69.68	OOM	72.66
	NAG	31.68	40.38	74.72	46.17	88.66	86.75	58.94	77.24	70.99	93.94	87.26	49.89	25.80	63.30	82.40	61.25
	Specformer	30.74	42.29	70.33	47.69	88.05	82.49	54.00	54.32	53.77	89.22	85.58	45.77	31.44	63.26	71.82	68.90
	Exphormer	36.62	59.36	68.34	65.63	89.22	78.17	68.19	61.80	57.31	94.25	86.97	76.31	42.31	79.47	83.84	80.39
	SGFormer	31.06	42.37	69.91	44.61	85.40	81.41	55.19	55.19	52.53	87.93	85.07	42.80	32.86	61.61	67.03	63.86
	VCR	26.41	27.28	71.15	28.13	82.45	85.41	44.89	61.64	58.32	88.45	83.25	44.66	22.48	54.68	78.46	51.33
	PolyFormer	32.12	46.83	72.13	48.91	89.60	83.51	59.02	58.11	56.15	91.93	86.82	47.29	34.42	64.38	73.18	69.77
Ours	Best-Validated	36.46	63.65	75.28	78.60	90.87	87.29	74.36	61.32	62.25	94.84	88.70	89.57	47.31	82.31	83.54	81.02
	CR-MAS	31.52	62.82	50.63	54.04	89.44	79.81	54.04	59.18	61.80	93.73	87.53	54.37	45.72	65.43	82.65	66.56
	Metric X	7 / 16	**14 / 16**	2 / 16	**11 / 16**	5 / 11	3 / 16	7 / 16	6 / 16	**13 / 16**	5 / 15	**13 / 13**	**12 / 16**	**14 / 14**	**9 / 16**	5 / 11	7 / 16
	Metric Y	-0.89	**11.64**	-17.12	**3.11**	**0.65**	**0.35**	-1.93	-1.65	**2.72**	**0.88**	**4.76**	**2.28**	**10.63**	-1.47	**3.01**	**0.42**
	CR-TAS	34.64	63.65	51.84	78.60	89.33	81.02	55.74	61.32	60.74	93.95	87.75	85.94	47.31	65.43	83.04	70.39
	Metric X	**11 / 16**	**14 / 16**	2 / 16	**16 / 16**	5 / 11	3 / 16	**8 / 16**	**12 / 16**	**13 / 16**	7 / 15	**13 / 13**	**15 / 16**	**14 / 14**	**9 / 16**	5 / 11	**9 / 16**
	Metric Y	**2.23**	**12.47**	-15.91	**27.67**	**0.54**	**1.56**	-0.23	**0.49**	**1.66**	**1.10**	**4.98**	**33.85**	**12.22**	-1.47	**3.40**	**4.25**

Metric X: # baselines outperformed by CR-GRAPHORMER (excl. OOMs, **bold** $\geq 50\%$); *Metric Y*: CR-GRAPHORMER accuracy minus baseline mean (**bold** ≥ 0)

From Table 6, we observe that CR-GRAPHORMER is competitive with the 16 baselines: CR-MAS and CR-TAS outperform at least half of them on $7/16$ and $11/16$ graphs ($6/12$ and $8/12$ in the target regimes), and exceed the mean baseline accuracy on 11 and 13 datasets, respectively. CR-MAS surpasses the best baseline on *squirrel* and *pubmed* by 0.31 and 0.20 percentage points; CR-TAS does so on *community*, *squirrel*, and *pubmed* by 5.40, 1.90, and 0.42 percentage points, respectively. CR-TAS is also the best validation-selected cascade model on *chameleon*, *community*, *cycle*, and *squirrel*, and achieves stronger overall performance than CR-MAS.

6 Conclusion

We introduce GRAPH CASCADES, a contagion-based mesoscopic rewiring procedure that constructs sparse auxiliary graphs for both message passing and attention. The rewiring improves structural resolution under reinforced multi-hop support. We provide theory connecting this to homophily and effective resistance, clarifying how rewiring can improve label agreement while favoring mesoscopically well-supported neighbors. The cascade-rewired neighborhoods also instantiate CR-GRAPHORMER, a sparse transformer whose token sequences are pre-selected by contagion reinforcement; this demonstrates that the auxiliary graph alone is sufficient to drive a sparse-attention GT at bounded sequence length. Across datasets, GRAPH CASCADES improves node classification for multiple GNN and GT backbones (with the largest improvements on heterophilic and moderate- to high-degree homophilic graphs). A regression study further demonstrates that performance aligns strongly with structural properties after rewiring, suggesting that the rewired structure carries the predictive signal.

Limitations. GRAPH CASCADES is less effective on highly regular, low-degree, or bottlenecked graphs, such as cycles, grids, and chain-like benchmarks including *Roman-empire*. In these settings, structural homogeneity and sparse connectivity limit the redundant short-path support needed for contagion-based rewiring. Theorem 4.8 formalizes this, while Appendix B.10 shows that fixed-threshold TAS-κ can add a nonlocal edge only when the target is already κ-edge-connected to the seed's closed neighborhood. Thus, when graphs are dominated by narrow cuts rather than cohesive mesoscopic regions, bottleneck-targeted rewiring methods may be preferable; combining these approaches is a natural direction for future work. As GRAPH CASCADES is topology-only by design, integrating cascade-based reinforcement with feature-aware edge selection is left to future work.

Acknowledgments. This research was supported in part by the Swedish Research Council project ExCLUS (2024 − 05603) and the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

References

Nimesh Agrawal, Anuj Kumar Sirohi, Sandeep Kumar, et al. No Prejudice! Fair Federated Graph Neural Networks for Personalized Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

Uri Alon and Eran Yahav. On the Bottleneck of Graph Neural Networks and its Practical Implications. *International Conference on Learning Representations*, 2021.

Adrián Arnaiz-Rodríguez, Ahmed Begga, Francisco Escolano, and Nuria M Oliver. DiffWire: Inductive Graph Rewiring via the Lovász Bound. *Proceedings of the First Learning on Graphs Conference*, 2022.

Hugo Attali, Davide Buscaldi, and Nathalie Pernelle. Delaunay graph: Addressing over-squashing and over-smoothing using delaunay triangulation. In *Forty-first International Conference on Machine Learning*, 2024.

Hugo Attali, Thomas Papastergiou, Nathalie Pernelle, and Fragkiskos D Malliaros. Dynamic triangulation-based graph rewiring for graph neural networks. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 87–97, 2025.

Federico Barbero, Ameya Velingker, Amin Saberi, Michael M Bronstein, and Francesco Di Giovanni. Locality-aware graph rewiring in gnns. In *The Twelfth International Conference on Learning Representations*, 2024.

Wendong Bi, Lun Du, Qiang Fu, Yanlin Wang, Shi Han, and Dongmei Zhang. Make Heterophilic Graphs Better Fit GNN: A Graph Rewiring Approach. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral Graph Neural Networks Meet Transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Aleksandar Bojchevski, Johannes Gasteiger, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2464–2473, 2020.

Damon Centola. *How Behavior Spreads: The Science of Complex Contagions*. Princeton University Press, 2018.

Damon Centola and Michael Macy. Complex Contagions and the Weakness of Long Ties. *American Journal of Sociology*, 113(3):702–734, 2007.

Meher Chaitanya, Kshitijaa Jaglan, and Ulrik Brandes. Adjacency Search Embeddings. *Transactions on Machine Learning Research*, 2025.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and Relieving the Over-Smoothing Problem for Graph Neural Networks from the Topological View. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2020.

Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-Aware Transformer for Graph Representation Learning. In *International conference on machine learning*. PMLR, 2022.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A Tokenized Graph Transformer for Node Classification in Large Graphs. In *The International Conference on Learning Representations*, 2023.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy Colwell, and Adrian Weller. Rethinking Attention with Performers. In *International Conference on Learning Representations*, 2021.

Ailin Deng and Bryan Hooi. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021.

Vijay Prakash Dwivedi and Xavier Bresson. A Generalization of Transformer Networks to Graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking Graph Neural Networks. *Journal of Machine Learning Research*, pages 1–48, 2023.

Wenzheng Feng, Yuxiao Dong, Tinglin Huang, Ziqi Yin, Xu Cheng, Evgeny Kharlamov, and Jie Tang. Grand+: Scalable graph random neural networks. In *Proceedings of the ACM Web Conference 2022*, pages 3248–3258, 2022.

Lukas Fesser and Melanie Weber. Mitigating Over-Smoothing and Over-Squashing using Augmentations of Forman-Ricci Curvature. In *Learning on Graphs Conference*, pages 19–1. PMLR, 2024.

Dongqi Fu, Zhigang Hua, Yan Xie, Jin Fang, Si Zhang, Kaan Sancak, Hao Wu, Andrey Malevich, Jingrui He, and Bo Long. VCR-Graphormer: A Mini-Batch Graph Transformer via Virtual Connections. In *The International Conference on Learning Representations*, 2024.

Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.

Mark Granovetter. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6): 1420–1443, 1978.

Florian Grötschla, Jiaqing Xie, and Roger Wattenhofer. Benchmarking Positional Encodings for GNNs and Graph Transformers. *arXiv preprint arXiv:2411.12732*, 2024.

Jiayan Guo, Lun Du, Wendong Bi, Qiang Fu, Xiaojun Ma, Xu Chen, Shi Han, Dongmei Zhang, and Yan Zhang. Homophily-oriented Heterogeneous Graph Rewiring. In *Proceedings of the ACM web conference 2023*, pages 511–522, 2023.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems*, 30, 2017.

Kedar Karhadkar, Pradeep Kr. Banerjee, and Guido Montufar. FoSR: First-order spectral rewiring for addressing oversquashing in GNNs. In *The International Conference on Learning Representations*, 2023.

Jinwoo Kim, Tien Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure Transformers are Powerful Graph Learners. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C Bayan Bruss, and Tom Goldstein. GOAT: A Global Transformer on Large-Scale Graphs. In *International Conference on Machine Learning*, pages 17375–17390. PMLR, 2023.

Weiyang Kong, Ziyu Guo, and Yubao Liu. Spatio-Temporal Pivotal Graph Neural Networks for Traffic Flow Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking Graph Transformers with Spectral Attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In *International conference on machine learning*, pages 13242–13256. PMLR, 2022.

Langzhang Liang, Sunwoo Kim, Kijung Shin, Zenglin Xu, Shirui Pan, and Yuan Qi. Sign is not a remedy: Multiset-to-multiset message passing for learning on heterophilic graphs. *arXiv preprint arXiv:2405.20652*, 2024.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and Basis Invariant Networks for Spectral Graph Representation Learning. *The International Conference on Learning Representations*, 2023.

Jonas Linkerhägner, Cheng Shi, and Ivan Dokmanić. Joint graph rewiring and feature denoising via spectral resonance. In *The Thirteenth International Conference on Learning Representations*, 2025.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375, 2022.

Jiahong Ma, Mingguo He, and Zhewei Wei. Polyformer: Scalable Node-Wise Filters via Polynomial Graph Transformer. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2118–2129, 2024.

Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding Graph Structure in Transformers. *arXiv preprint arXiv:2106.05667*, 2021.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2019.

Khang Nguyen, Nong Minh Hieu, Vinh Duc Nguyen, Nhat Ho, Stanley Osher, and Tan Minh Nguyen. Revisiting Over-Smoothing and Over-Squashing using Ollivier-Ricci Curvature. In *International Conference on Machine Learning*, pages 25956–25979. PMLR, 2023.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, pages 14501–14515, 2022.

Celia Rubio-Madrigal, Adarsh Jamadandi, and Rebekka Burkholz. Gnns getting comfy: Community and feature similarity guided rewiring. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse Transformers for Graphs. In *International Conference on Machine Learning*, pages 31613–31632. PMLR, 2023.

Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. Ordered gnn: Ordering message passing to deal with heterophily and over-smoothing. *arXiv preprint arXiv:2302.01524*, 2023.

Daniel A. Spielman and Nikhil Srivastava. Graph Sparsification by Effective Resistances. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 563–568, 2008.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. NodeFormer: A Scalable Graph Structure Learning Transformer for Node Classification. *Advances in Neural Information Processing Systems*, pages 27387–27401, 2022.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. SGFormer: Simplifying and Empowering Transformers for Large-Graph Representations. *Advances in Neural Information Processing Systems*, pages 64753–64773, 2023.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1287–1292. IEEE, 2022.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do Transformers Really Perform Badly for Graph Representation? *Advances in Neural Information Processing Systems*, pages 28877–28888, 2021.

Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the Expressive Power of GNNs via Graph Biconnectivity. In *The International Conference on Learning Representations*, 2023.

Jianan Zhao, Chaozhuo Li, Qianlong Wen, Yiqi Wang, Yuming Liu, Hao Sun, Xing Xie, and Yanfang Ye. Gophormer: Ego-Graph Transformer for Node Classification. *arXiv preprint arXiv:2110.13094*, 2021.

Zhuonan Zheng, Yuanchen Bei, Zhiyao Zhou, Sheng Zhou, Yao Ma, Ming Gu, Hongjia Xu, Jiawei Chen, and Jiajun Bu. Understanding and enhancing message passing on heterophilic graphs via compatibility matrix. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zijie Zhou, Zhaoqi Lu, Xuekai Wei, Rongqin Chen, Shenghui Zhang, Pak Lon Ip, et al. Tokenphormer: Structure-Aware Multi-Token Graph Transformer for Node Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13428–13436, 2025.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.

APPENDIX

A Preliminaries and Related Work

A.1 Preliminaries

Graph Neural Networks (GNNs). Let $\mathbf{H} \in \mathbb{R}^{n \times d}$ be the node features of G, with row $\mathbf{h}_i \in \mathbb{R}^d$ for node i. A GNN processes $\mathbf{H}$ through message-passing layers that respect graph sparsity, updating each node's embedding by aggregating over its one-hop neighborhood:

$$\mathbf{h}'_i = \phi\left(\mathbf{h}_i, \bigoplus_{j \in N(i)} \psi(\mathbf{h}_i, \mathbf{h}_j)\right),$$

where ψ and ϕ are learnable and $\bigoplus$ is a permutation-invariant aggregator. Stacking L layers expands the receptive field to L-hop neighborhoods at the cost of oversmoothing, oversquashing, and weak long-range interactions [Alon and Yahav, 2021]. Sparse variants restrict message passing to a subset of neighbors, equivalent to running the same update on a rewired graph with neighborhoods $N^\diamond(i)$ in place of $N(i)$.

Graph Transformers (GTs). GTs adopt the standard transformer block: an attention layer followed by a feed-forward operation. Dense variants [Ying et al., 2021, Mialon et al., 2021, Kreuzer et al., 2021, Zhao et al., 2021] treat each node as a token attending to every other node, encoding graph topology through positional encodings such as shortest-path embeddings [Ying et al., 2021, Zhang et al., 2023] or Laplacian eigenvectors [Dwivedi and Bresson, 2020]. The single-head self-attention update over a node set $N^\diamond(i)$ is

$$\mathbf{h}'_i = \sum_{j \in N^\diamond(i)} \alpha_{ij}\, \mathbf{v}_j, \qquad \alpha_{ij} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d'})}{\sum_{r \in N^\diamond(i)} \exp(\mathbf{q}_i^\top \mathbf{k}_r / \sqrt{d'})},$$

where $\mathbf{Q} = \mathbf{H}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{H}\mathbf{W}_K$, $\mathbf{V} = \mathbf{H}\mathbf{W}_V$ are learnable projections with $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d'}$. Setting $N^\diamond(i) = N(i)$ recovers one-hop attention as in GATs [Veličković et al., 2018]. Scalable sparse variants instead define $N^\diamond(\cdot)$ via diffusion- or range-limited neighborhoods (NAGphormer [Chen et al., 2023], VCR-Graphormer [Fu et al., 2024]) or fixed expander/landmark patterns (Exphormer [Shirzad et al., 2023]).

These architectures rely on either strictly local neighborhoods or sparsified approximations of global interactions, and do not explicitly encode intermediate-scale structure. Our proposed GRAPH CASCADES provides a structural augmentation that induces mesoscopic connectivity patterns, enabling efficient modeling of interactions beyond one-hop neighborhoods in both GNN and GT architectures.

A.2 Extended Related Work

Existing research on GTs can be broadly classified into three categories: (i) GT architectures, (ii) positional encodings for graphs, and (iii) graph rewiring techniques.

GT Architectures. GT models adapt the self-attention mechanism of the original transformer [Vaswani et al., 2017] to graph-structured data. The GT network [Dwivedi and Bresson, 2020] applies dense self-attention across all $\mathcal{O}(n^2)$ node pairs, achieving strong performance on several benchmarks. Subsequent architectures introduce graph-specific inductive biases into the attention mechanism: Graphormer [Ying et al., 2021] encodes node centrality and pairwise shortest-path distances; GraphiT [Mialon et al., 2021] employs kernelized relative encodings and explicit subpath features; and SAN [Kreuzer et al., 2021] and Gophormer [Zhao et al., 2021] incorporate edge and node attributes. Although these models achieve strong performance across domains such as recommendation, question answering, and bioinformatics, their fully connected attention mechanisms incur quadratic computational cost.

To address scalability, a growing body of work focuses on lightweight and sparse-attention GT variants. Exphormer [Shirzad et al., 2023] imposes expander-graph sparsity and GOAT [Kong et al., 2023] projects node features into lower-dimensional spaces. NodeFormer [Wu et al., 2022] introduces topology-aware relational biases, achieving provably linear complexity in the number

of nodes. A parallel line of work embeds graphs to confine attention to compact, information-rich subsets through sparse tokenization schemes. TokenGT [Kim et al., 2022] represents nodes and edges as independent tokens with structural positional encodings, achieving competitive or superior performance to GNNs on molecular benchmarks. Tokenphormer [Zhou et al., 2025] extends this idea by generating mixed random-walk, hop, and global SGPM tokens for each node, enabling applicability to both homogeneous and heterogeneous graphs; however, its multi-token aggregation lacks an explicit heterophily-aware mechanism and may introduce label noise when neighboring nodes exhibit dissimilar labels or attributes. PolyFormer [Ma et al., 2024] introduces PolyAttn, an attention-based node-wise polynomial graph filter that avoids the expensive positional encodings used by prior node-wise spectral methods, improving scalability. Building on PolyAttn, PolyFormer attends over per-node polynomial "tokens" to efficiently capture spectral information and reports strong performance on both homophilic and heterophilic node-level benchmarks, scaling to very large graphs. SGFormer [Wu et al., 2023] argues that a deep, multi-head GT is not necessary for strong performance: a single-layer global attention model can be surprisingly competitive on node property prediction while being much cheaper to run. It removes positional encodings and other preprocessing from its design and scales linearly with the number of nodes. Specformer [Bo et al., 2023] is a spectral GNN that replaces traditional scalar spectral filters with a learnable set-to-set spectral filter by encoding all Laplacian eigenvalues and applying self-attention in the spectral domain. It also introduces a learnable-basis decoder to enable nonlocal graph convolution, while remaining permutation-equivariant, and reports strong gains on both node- and graph-level benchmarks. VCR-Graphormer [Fu et al., 2024] explicitly targets heterophily by introducing virtual connections and PPR-based token lists that encode both local and global dependencies while maintaining linear complexity per layer. Nevertheless, its reliance on virtual connections and structural partitioning increases preprocessing overhead and complicates scalability. *In contrast, our label-free mesoscopic rewiring captures both structural and heterophily-driven relationships without requiring virtual connections or graph partitioning, yielding improved performance on heterophilic graphs, and maintaining competitive accuracy on homophilic ones.*

Positional Encodings for Graphs. Since graphs lack an intrinsic notion of order, GTs must rely on positional encodings to inject structural information. Laplacian-based encodings (LapPE) [Dwivedi and Bresson, 2020], full-spectrum methods [Kreuzer et al., 2021], and sign-invariant variants such as SignNet [Lim et al., 2023] assign each node absolute coordinates in a spectral space, but can be sensitive to eigenvector permutations. Random-walk and diffusion-based encodings (RWSE, RWDiffusion, RRWP) [Dwivedi et al., 2023, Grötschla et al., 2024] capture multi-scale proximity patterns, while shortest-path or distance encodings underpin Graphormer's attention biases [Ying et al., 2021]. Weisfeiler–Lehman subtree encodings (WL-PE) [Morris et al., 2019] and edge-aware learnable schemes such as PureGT's relative positional encodings [Kim et al., 2022] further enrich positional signals. Systematic studies [Rampášek et al., 2022] demonstrate that combining complementary positional encodings (e.g., Laplacian and shortest-path encodings) yields robust gains across heterogeneous benchmarks. Overall, well-designed positional encodings are now recognized as a primary driver of GT performance, enabling discrimination of node roles, distances, and higher-order structural motifs.

Graph Rewiring Techniques. Beyond architectural design, graph rewiring has emerged as an effective strategy for improving learning on challenging graph structures. HDHGR [Guo et al., 2023] and DHGR [Bi et al., 2024] rely on (pseudo-)labels to add homophilic edges and remove heterophilic ones, which can discard informative cross-class links and make performance dependent on label quality and availability. *In contrast, our mesoscopic rewiring approach is entirely label-free. It connects node pairs reinforced by multi-path short walks (Theorem 4.8) while pruning weak or noisy edges, yielding consistent gains on heterophilic graphs (Proposition 4.5 and Theorem 4.7) and remains competitive on many homophilic graphs, with predictable failures on low-degree regular structures. Our method integrates seamlessly with GTs using sparse attention or as a standalone GT, without introducing virtual edges or requiring graph partitioning.* Curvature-based approaches [Topping et al., 2022, Nguyen et al., 2023, Fesser and Weber, 2024] and differentiable or spectral rewiring methods [Arnaiz-Rodríguez et al., 2022, Karhadkar et al., 2023] typically aim to increase algebraic connectivity to facilitate information flow. *By contrast, our approach does not raise global connectivity or introduce shortcuts between weakly connected regions; instead, it emphasizes walk-reinforced* mesoscopic *ties that better capture both structural and heterophily-driven relationships.* PPRGo [Bojchevski et al., 2020] is a scalable GNN that decouples local feature transformation from graph propagation and then propagates predictions using precomputed sparse approximations of personalized PageRank (PPR),

avoiding expensive multi-hop message passing or per-step power iteration while keeping long-range information. It is not "rewiring" in the strict topology-editing sense (i.e., explicitly adding/removing edges in the original graph), but it does effectively replace the adjacency-based propagation with a sparsified PPR diffusion operator which is equivalent to working on an implicit new weighted neighborhood graph defined by top-PPR mass—so it is reasonable to describe it as an implicit diffusion-based rewiring mechanism. GRAND+ [Feng et al., 2022] is a scalable version of GRAND that precomputes a (generalized) mixed-order propagation matrix using a generalized forward push (GFPush) algorithm, so it can do GRAND-style random propagation graph augmentations in mini-batches, and it adds a confidence-aware consistency loss to improve generalization. It is not a formal method, i.e., it does not edit the original edge set, but because it replaces adjacency-based propagation with a learned/selected propagation matrix (a diffusion operator), it effectively induces an implicit weighted "rewired" neighborhood structure for propagation/augmentation.

Positioning Relative to Recent Rewiring Directions. Recent rewiring methods can be organized along two axes: the structural objective (bottleneck mitigation vs. neighborhood realignment) and the information used (purely structural vs. feature- or label-aware). Curvature-based [Topping et al., 2022, Nguyen et al., 2023, Fesser and Weber, 2024] and spectral [Karhadkar et al., 2023, Arnaiz-Rodríguez et al., 2022] methods are structural and target spectral bottlenecks to alleviate oversquashing. Locality-aware rewiring [Barbero et al., 2024] preserves local geometry while adding shortcuts. Feature-aware methods take a complementary route: Delaunay-based approaches [Attali et al., 2024, 2025] replace the graph via feature-space triangulation, joint spectral-resonance methods [Linkerhägner et al., 2025] co-optimize feature denoising and rewiring, and community–feature similarity methods [Rubio-Madrigal et al., 2025] combine structural and feature signals. These methods use strictly more information than the original graph topology and are most appropriate when features are reliable and informative.

GRAPH CASCADES occupies a distinct position on both axes. It is purely structural and label-free, like FOSR [Karhadkar et al., 2023] and SDRF [Topping et al., 2022], but its objective is neighborhood realignment via reinforced multi-hop co-activation rather than bottleneck mitigation. Theorem 4.8 formalizes this: cascade activations are supported by κ edge-disjoint short paths, so the rewiring concentrates on cohesive mesoscopic regions rather than spanning weak cuts. Empirically, this distinction is reflected in where cascades help most—heterophilic graphs whose label structure is carried by reinforced multi-hop neighborhoods—and where they do not (chain-like or grid-like structures, where bottleneck-targeted methods remain more appropriate). GRAPH CASCADES is therefore best understood as complementary to existing rewiring directions rather than as a replacement for any of them.

B Proofs of Theoretical Results

In this section, we provide proofs of some theoretical results introduced in the main body.

B.1 Proof of Proposition 4.5

By Bayes' rule,

$$h_G = \Pr[\mathcal{S} \mid \mathcal{E}] = \frac{\Pr[\mathcal{E} \mid \mathcal{S}] \Pr[\mathcal{S}]}{\Pr[\mathcal{E}]},$$

and

$$h_{\mathcal{F}} = \Pr[\mathcal{S} \mid \mathcal{F}] = \frac{\Pr[\mathcal{F} \mid \mathcal{S}] \Pr[\mathcal{S}]}{\Pr[\mathcal{F}]}.$$

By Assumption 4.3,

$$\Pr[\mathcal{F} \mid \mathcal{S}] \geq \Pr[\mathcal{E} \mid \mathcal{S}].$$

Therefore,

$$h_{\mathcal{F}} \geq \frac{\Pr[\mathcal{E} \mid \mathcal{S}] \Pr[\mathcal{S}]}{\Pr[\mathcal{F}]} = h_G \frac{\Pr[\mathcal{E}]}{\Pr[\mathcal{F}]}.$$

If Assumption 4.4 also holds, then

$$\frac{\Pr[\mathcal{E}]}{\Pr[\mathcal{F}]} \geq 1,$$

and so

$$h_{\mathcal{F}} \geq h_G.$$

If Assumption 4.3 is strict, then the first inequality above is strict, so $h_{\mathcal{F}} > h_G$ whenever Assumption 4.4 holds. If Assumption 4.4 is strict and $h_G > 0$, then

$$h_G \frac{\Pr[\mathcal{E}]}{\Pr[\mathcal{F}]} > h_G,$$

and therefore, $h_{\mathcal{F}} > h_G$.

B.2 Proof of Theorem 4.7

We first state the lemma used in the proof for completeness.

Lemma B.1 (Chernoff/Bernstein Bounds for Poisson-Binomial Sums)**.** *Let $X = \sum_i X_i$, where the X_i are independent Bernoulli random variables with mean $\mu = \mathbb{E}[X]$. Then, for $0 \leq t \leq \mu$,*

$$\mathrm{P}[X - \mu \leq -t] \leq \exp\left(-\frac{t^2}{2\mu}\right),$$

and, for $t \geq 0$,

$$\mathrm{P}[X - \mu \geq t] \leq \exp\left(-\frac{t^2}{2\mu + 2t/3}\right).$$

Proof. The lower-tail inequality follows from the multiplicative Chernoff bound $\mathrm{P}[X \leq (1-\delta)\mu] \leq \exp(-\delta^2\mu/2)$ with $t = \delta\mu$. The upper-tail inequality follows from Bernstein's inequality using $\mathrm{Var}(X) \leq \mu$ and $|X_i - \mathbb{E}[X]_i| \leq 1$. □

We prove Theorem 4.7 in five steps.

Step 1: Population Identity. Fix two distinct nodes u, v with $y_u = a$ and $y_v = b$. By definition,

$$s_2(u,v) = |N(u) \cap N(v)| = \sum_{x \in V \setminus \{u,v\}} A_{ux}A_{vx}.$$

Conditional on the labels, the summands are independent across x, because different x's involve disjoint unordered edge variables. If $y_x = c$, then

$$\begin{aligned}\mathbb{E}[A_{ux}A_{vx} \mid y_u = a, y_v = b, y_x = c] &= \mathbb{E}[A_{ux} \mid y_u = a, y_x = c]\,\mathbb{E}[A_{vx} \mid y_v = b, y_x = c] \\ &= B_{ac}B_{bc}.\end{aligned}$$

Therefore,

$$\begin{aligned}\mu_{ab}^{(n)} &= \sum_{c=1}^{C}\big(n_c - \mathbb{I}(c=a) - \mathbb{I}(c=b)\big)B_{ac}B_{bc} = n\sum_{c=1}^{C}\pi_c B_{ac}B_{bc} + O(1) \\ &= n(BD_\pi B^\top)_{ab} + O(1).\end{aligned}$$

Equivalently,

$$\frac{1}{n}\mu_{ab}^{(n)} = (BD_\pi B^\top)_{ab} + O(1/n).$$

Define the finite-sample profile margin and the maximum two-hop mean by

$$\gamma_n := \min_{a \in [C]}\left(\mu_{aa}^{(n)} - \max_{b \neq a}\mu_{ab}^{(n)}\right), \qquad \mu_{\max} := \max_{a,b \in [C]}\mu_{ab}^{(n)}.$$

By the population identity and $\Delta_2 > 0$,

$$\gamma_n = n\Delta_2 + O(1), \qquad \mu_{\max} = O(n).$$

Thus, for all sufficiently large n,

$$\gamma_n^2 \geq 48\mu_{\max}\log n, \qquad \gamma_n \geq 16\log n. \tag{3}$$

We assume from now on that n is large enough for (3) to hold.

Step 2: Per-Class Midpoint Thresholds. For each class $a \in [C]$, define the midpoint threshold

$$\kappa_a^\star := \frac{\mu_{aa}^{(n)} + \max_{b \neq a} \mu_{ab}^{(n)}}{2}.$$

Then

$$\mu_{aa}^{(n)} - \kappa_a^\star = \frac{\mu_{aa}^{(n)} - \max_{b \neq a} \mu_{ab}^{(n)}}{2} \geq \frac{\gamma_n}{2}.$$

For every $b \neq a$,

$$\kappa_a^\star - \mu_{ab}^{(n)} \geq \kappa_a^\star - \max_{b' \neq a} \mu_{ab'}^{(n)} = \frac{\mu_{aa}^{(n)} - \max_{b' \neq a} \mu_{ab'}^{(n)}}{2} \geq \frac{\gamma_n}{2}.$$

Thus, one threshold $\kappa_a^\star$ separates the same-class population mean from all different-class population means for seed class a, with margin at least $\gamma_n/2$ on both sides.

Step 3: Concentration Around the Midpoint. Fix distinct u, v with $y_u = a$ and $y_v = b$. Then $s_2(u, v)$ is a sum of independent Bernoulli variables with mean $\mu_{ab}^{(n)} \leq \mu_{\max}$.

Same-class case. Assume $b = a$. Let

$$t_{\text{same}} := \mu_{aa}^{(n)} - \kappa_a^\star \geq \gamma_n/2.$$

Since $\kappa_a^\star \geq 0$, we have $t_{\text{same}} \leq \mu_{aa}^{(n)}$, so Lemma B.1 applies:

$$\mathrm{P}\left[s_2(u, v) \leq \kappa_a^\star \mid y\right] \leq \exp\left(-\frac{t_{\text{same}}^2}{2\mu_{aa}^{(n)}}\right) \leq \exp\left(-\frac{\gamma_n^2}{8\mu_{\max}}\right).$$

By (3), the exponent is at most $-6 \log n$ and therefore

$$\mathrm{P}\left[s_2(u, v) \leq \kappa_a^\star \mid y\right] \leq n^{-6} \leq n^{-4}.$$

Different-class case. Assume $b \neq a$. Let

$$t_{\text{diff}} := \kappa_a^\star - \mu_{ab}^{(n)} \geq \gamma_n/2.$$

By Lemma B.1,

$$\mathrm{P}\left[s_2(u, v) \geq \kappa_a^\star \mid y\right] \leq \exp\left(-\frac{t_{\text{diff}}^2}{2\mu_{ab}^{(n)} + 2t_{\text{diff}}/3}\right).$$

For fixed $\mu \geq 0$, the function

$$\phi_\mu(t) := \frac{t^2}{2\mu + 2t/3}, \qquad t > 0,$$

is increasing, since

$$\phi'_\mu(t) = \frac{4\mu t + (2/3)t^2}{(2\mu + 2t/3)^2} > 0.$$

Using $t_{\text{diff}} \geq \gamma_n/2$ and $\mu_{ab}^{(n)} \leq \mu_{\max}$,

$$\frac{t_{\text{diff}}^2}{2\mu_{ab}^{(n)} + 2t_{\text{diff}}/3} \geq \frac{(\gamma_n/2)^2}{2\mu_{ab}^{(n)} + \gamma_n/3} = \frac{\gamma_n^2}{8\mu_{ab}^{(n)} + (4/3)\gamma_n} \geq \frac{\gamma_n^2}{8\mu_{\max} + (4/3)\gamma_n}.$$

We show that this last quantity is at least $4 \log n$. It is enough to prove

$$\gamma_n^2 \geq 4 \log n \big(8\mu_{\max} + (4/3)\gamma_n\big) = 32\mu_{\max} \log n + (16/3)\gamma_n \log n.$$

Since $\gamma_n \geq 16 \log n$,

$$(16/3)\gamma_n \log n \leq \gamma_n^2/3.$$

Also $\gamma_n^2 \geq 48\mu_{\max} \log n$ implies

$$32\mu_{\max} \log n \leq (2/3)\gamma_n^2.$$

Therefore,

$$32\mu_{\max} \log n + (16/3)\gamma_n \log n \leq (2/3)\gamma_n^2 + (1/3)\gamma_n^2 = \gamma_n^2.$$

Hence,

$$\mathrm{P}\left[s_2(u, v) \geq \kappa_a^\star \mid y\right] \leq n^{-4}.$$

Step 4: Union Bound Over Ordered Pairs. Define the favorable event

$$\mathcal{F} := \bigcap_{u,v\in V:\, u\neq v,\ y_u=y_v} \{s_2(u,v) > \kappa^\star_{y_u}\} \cap \bigcap_{u,v\in V:\, u\neq v,\ y_u\neq y_v} \{s_2(u,v) < \kappa^\star_{y_u}\}.$$

There are at most $n(n-1) \le n^2$ ordered pairs (u,v), and each corresponding failure event has probability at most n^{-4}. Thus,

$$\mathrm{P}\left[\mathcal{F}^c \mid y\right] \le n^2 n^{-4} = n^{-2} \le 2n^{-2}.$$

Therefore, $\mathrm{P}\left[\mathcal{F} \mid y\right] \ge 1 - 2n^{-2}$.

Step 5: Uniform Separation and Top-k Homophily. On $\mathcal{F}$, fix any seed u with $y_u = a$. For every same-class candidate v and every different-class candidate w,

$$s_2(u,v) > \kappa^\star_a > s_2(u,w).$$

Hence, every same-class candidate has strictly larger two-hop score than every different-class candidate relative to seed u.

If $k \le \min_a(n_a - 1)$, every seed u has at least k other nodes in its own class. Hence, the directed top-k rule from u selects only same-label nodes. Symmetrization cannot create a cross-label edge, because an undirected edge $\{u,v\}$ appears only if at least one of the directed edges $u \to v$ or $v \to u$ was selected, and every selected directed edge is same-label. Consequently, $h_{G_2^\star(k)} = 1$.

B.2.1 Edge Homophily of the Original SBM

Corollary B.2 (Limiting Edge Homophily of the Original SBM)**.** *Under the SBM setting above, assume* $\sum_{a,b=1}^{C} \pi_a \pi_b B_{ab} > 0$*. Then the edge homophily of the original graph satisfies the following:*

$$h_G \xrightarrow{p} \frac{\sum_{a=1}^{C} \pi_a^2 B_{aa}}{\sum_{a,b=1}^{C} \pi_a \pi_b B_{ab}},$$

where $\xrightarrow{p}$ *denotes convergence in probability. In particular, the limit is strictly below* $1/2$ *whenever* $\sum_{a=1}^{C} \pi_a^2 B_{aa} < \sum_{a\neq b} \pi_a \pi_b B_{ab}$*.*

Proof. Let M_{same} denote the number of same-label edges and M the total number of edges, so that $h_G = M_{\mathrm{same}}/M$. By independence of edges in the SBM,

$$\mathbb{E}[M_{\mathrm{same}}] = \sum_{a=1}^{C} \binom{n_a}{2} B_{aa} = \frac{n^2}{2} \sum_{a=1}^{C} \pi_a^2 B_{aa} + O(n), \tag{4}$$

and

$$\mathbb{E}[M] = \sum_{a=1}^{C} \binom{n_a}{2} B_{aa} + \sum_{a<b} n_a n_b B_{ab} = \frac{n^2}{2} \sum_{a,b=1}^{C} \pi_a \pi_b B_{ab} + O(n). \tag{5}$$

Both M_{same} and M are sums of independent Bernoulli random variables, so

$$\mathrm{Var}(M) \le \mathbb{E}[M], \qquad \mathrm{Var}(M_{\mathrm{same}}) \le \mathbb{E}[M_{\mathrm{same}}].$$

Since $\sum_{a,b} \pi_a \pi_b B_{ab} > 0$, (5) gives $\mathbb{E}[M] = \Theta(n^2)$, so Chebyshev's inequality implies $M/\mathbb{E}[M] \xrightarrow{p} 1$. If $\sum_a \pi_a^2 B_{aa} > 0$, the same argument applied to (4) gives $M_{\mathrm{same}}/\mathbb{E}[M_{\mathrm{same}}] \xrightarrow{p} 1$, and the continuous-mapping theorem yields

$$h_G \xrightarrow{p} \frac{\sum_{a=1}^{C} \pi_a^2 B_{aa}}{\sum_{a,b=1}^{C} \pi_a \pi_b B_{ab}}. \tag{6}$$

If $\sum_a \pi_a^2 B_{aa} = 0$, then $B_{aa} = 0$ for every a with $\pi_a > 0$, so $M_{\mathrm{same}} = 0$ deterministically and (6) still holds with limit zero. □

B.3 Learnability After Cascade Rewiring

Corollary B.3 (Learnability After Cascade Rewiring)**.** *Assume the setting and profile-margin condition of Theorem 4.7, and let $G_2^\star(k)$ be the symmetrized top-k two-hop cascade-rewired graph, with $k \le \min_{a\in[C]}(n_a - 1)$. Suppose, independently of the graph conditional on the labels, node features are generated by $X_i = \mu_{y_i} + \sigma\xi_i,\ \xi_i \sim \mathcal{N}(0, I_d)$, where $\mu_1, \dots, \mu_C \in \mathbb{R}^d$, $\sigma > 0$, and $\Delta := \min_{a\neq b} \|\mu_a - \mu_b\|_2 > 0$. For each node i, let $N_\star(i)$ denote its neighborhood in $G_2^\star(k)$, and define $Z_i := \frac{1}{q_i}\sum_{j\in\{i\}\cup N_\star(i)} X_j,\ q_i := |\{i\} \cup N_\star(i)|$. Consider the nearest-centroid affine linear classifier $\widehat{y}_i := \arg\max_{a\in[C]} \left\{\mu_a^\top Z_i - \frac{1}{2}\|\mu_a\|_2^2\right\}$. Then, with probability at least $1 - 2n^{-2} - n(C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right)$, all nodes are classified correctly. In particular, if $\frac{(k+1)\Delta^2}{8\sigma^2} \ge \log n + \log(C-1) + r_n$ for some $r_n \to \infty$, then $\Pr[\widehat{y}_i = y_i \text{ for all } i \in V] \to 1$. Moreover, the expected misclassification fraction satisfies $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \mathbf{1}\{\widehat{y}_i \neq y_i\}\right] \le 2n^{-2} + (C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right)$.*

Proof. We prove the result conditional on the label vector $y = (y_1, \dots, y_n)$. All probabilities below are over the randomness of the graph and the Gaussian features. The same bounds hold unconditionally if the labels are random and independent of the feature noise. Let

$$\mathcal{E}_{\text{hom}} := \{G_2^\star(k) \text{ is perfectly homophilic}\}.$$

By Theorem 4.7, under the stated profile-margin condition, $\Pr[\mathcal{E}_{\text{hom}}] \ge 1 - 2n^{-2}$. Equivalently, $\Pr[\mathcal{E}_{\text{hom}}^c] \le 2n^{-2}$. We first analyze the classifier on the event $\mathcal{E}_{\text{hom}}$. Fix a realization of $G_2^\star(k)$ for which $\mathcal{E}_{\text{hom}}$ holds. Since $G_2^\star(k)$ is perfectly homophilic, every edge of $G_2^\star(k)$ connects two vertices with the same label. Therefore, for every node i, every vertex in $\{i\} \cup N_\star(i)$ has label y_i.

Moreover, because $G_2^\star(k)$ is the symmetrization of a directed top-k graph, node i is adjacent in the symmetrized graph to at least the k vertices selected by the directed top-k rule from i. Hence,

$$q_i = |\{i\} \cup N_\star(i)| \ge k + 1.$$

Fix a node $i \in V$ and let $a := y_i$. On $\mathcal{E}_{\text{hom}}$, every $j \in \{i\} \cup N_\star(i)$ has label a. Since the features are independent conditional on the labels, and are independent of the graph conditional on the labels, we have, conditional on $G_2^\star(k)$, $\mathcal{E}_{\text{hom}}$, and the labels,

$$X_j \sim \mathcal{N}(\mu_a, \sigma^2 I_d) \qquad \text{independently for all } j \in \{i\} \cup N_\star(i).$$

Therefore,

$$Z_i = \frac{1}{q_i}\sum_{j\in\{i\}\cup N_\star(i)} X_j \sim \mathcal{N}\left(\mu_a, \frac{\sigma^2}{q_i} I_d\right).$$

Equivalently, we may write

$$Z_i = \mu_a + \eta_i, \qquad \eta_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{q_i} I_d\right).$$

The classifier is the nearest-centroid classifier because

$$\arg\max_{c\in[C]} \left\{\mu_c^\top Z_i - \frac{1}{2}\|\mu_c\|_2^2\right\} = \arg\min_{c\in[C]} \|Z_i - \mu_c\|_2^2.$$

As

$$\|Z_i - \mu_c\|_2^2 = \|Z_i\|_2^2 - 2\mu_c^\top Z_i + \|\mu_c\|_2^2,$$

the term $\|Z_i\|_2^2$ does not depend on c.

Now fix an incorrect class $b \neq a$. The classifier can prefer b over the true class a only if

$$\|Z_i - \mu_b\|_2^2 \le \|Z_i - \mu_a\|_2^2.$$

Substituting $Z_i = \mu_a + \eta_i$, and defining

$$\delta_{ab} := \mu_a - \mu_b,$$

this event becomes

$$\|\eta_i + \delta_{ab}\|_2^2 \le \|\eta_i\|_2^2.$$

Expanding both sides gives

$$\|\eta_i\|_2^2 + 2\delta_{ab}^\top \eta_i + \|\delta_{ab}\|_2^2 \le \|\eta_i\|_2^2,$$

or equivalently

$$\delta_{ab}^\top \eta_i \le -\frac{1}{2}\|\delta_{ab}\|_2^2.$$

Since

$$\eta_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{q_i} I_d\right),$$

the one-dimensional projection satisfies

$$\delta_{ab}^\top \eta_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{q_i}\|\delta_{ab}\|_2^2\right).$$

Using the standard Gaussian tail bound

$$\Pr[W \le -t] \le \exp\left(-\frac{t^2}{2s^2}\right), \qquad W \sim \mathcal{N}(0, s^2),$$

with

$$t = \frac{1}{2}\|\delta_{ab}\|_2^2, \qquad s^2 = \frac{\sigma^2}{q_i}\|\delta_{ab}\|_2^2,$$

we obtain

$$\Pr\left[\|Z_i - \mu_b\|_2^2 \le \|Z_i - \mu_a\|_2^2 \mid G_2^\star(k), \mathcal{E}_{\mathrm{hom}}, y\right] \le \exp\left(-\frac{q_i\|\delta_{ab}\|_2^2}{8\sigma^2}\right).$$

By the definitions of q_i and Δ,

$$q_i \ge k+1, \qquad \|\delta_{ab}\|_2 = \|\mu_a - \mu_b\|_2 \ge \Delta.$$

Hence,

$$\Pr\left[\|Z_i - \mu_b\|_2^2 \le \|Z_i - \mu_a\|_2^2 \mid G_2^\star(k), \mathcal{E}_{\mathrm{hom}}, y\right] \le \exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).$$

Taking a union bound over all incorrect classes $b \ne a$, we get

$$\Pr[\widehat{y}_i \ne y_i \mid G_2^\star(k), \mathcal{E}_{\mathrm{hom}}, y] \le (C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).$$

This bound is uniform in the realization of $G_2^\star(k)$ on $\mathcal{E}_{\mathrm{hom}}$. Therefore,

$$\Pr[\widehat{y}_i \ne y_i \mid \mathcal{E}_{\mathrm{hom}}, y] \le (C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).$$

Next, taking a union bound over all n vertices gives

$$\Pr[\exists i \in V : \widehat{y}_i \ne y_i \mid \mathcal{E}_{\mathrm{hom}}, y] \le n(C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).$$

Therefore,

$$\begin{aligned}\Pr[\exists i \in V : \widehat{y}_i \ne y_i] &\le \Pr[\mathcal{E}_{\mathrm{hom}}^c] + \Pr[\exists i \in V : \widehat{y}_i \ne y_i \mid \mathcal{E}_{\mathrm{hom}}, y] \\ &\le 2n^{-2} + n(C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).\end{aligned}$$

Equivalently, with probability at least

$$1 - 2n^{-2} - n(C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right),$$

we have

$$\widehat{y}_i = y_i \qquad \text{for all } i \in V.$$

Now suppose

$$\frac{(k+1)\Delta^2}{8\sigma^2} \geq \log n + \log(C-1) + r_n,$$

where $r_n \to \infty$. Then

$$n(C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right) \leq n(C-1)\exp\left(-\log n - \log(C-1) - r_n\right) = e^{-r_n} \to 0.$$

Since also $2n^{-2} \to 0$, we conclude that

$$\Pr\left[\widehat{y}_i = y_i \text{ for all } i \in V\right] \to 1.$$

It remains to prove the expected misclassification fraction bound. Define

$$L_n := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\widehat{y}_i \neq y_i\}.$$

Since $0 \leq L_n \leq 1$,

$$\mathbb{E}[L_n] = \mathbb{E}[L_n \mathbf{1}_{\mathcal{E}_{\text{hom}}}] + \mathbb{E}[L_n \mathbf{1}_{\mathcal{E}_{\text{hom}}^c}] \leq \mathbb{E}[L_n \mathbf{1}_{\mathcal{E}_{\text{hom}}}] + \Pr\left[(\right] \mathcal{E}_{\text{hom}}^c).$$

Using the single-node error bound on $\mathcal{E}_{\text{hom}}$,

$$\begin{aligned}
\mathbb{E}[L_n \mathbf{1}_{\mathcal{E}_{\text{hom}}}] &= \frac{1}{n}\sum_{i=1}^{n} \Pr\left[\widehat{y}_i \neq y_i, \mathcal{E}_{\text{hom}}\right] \\
&\leq \frac{1}{n}\sum_{i=1}^{n} (C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right) \\
&= (C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).
\end{aligned}$$

Combining this with $\Pr\left[\mathcal{E}_{\text{hom}}^c\right] \leq 2n^{-2}$, we obtain

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\widehat{y}_i \neq y_i\}\right] \leq 2n^{-2} + (C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).$$

□

B.4 Extending Profile Homophily on SBM : $\ell \geq 2$

Lemma B.4 (Hoeffding Bound for Dependency Graphs)**.** *Let $X_1, \ldots, X_N \in [0,1]$ be random variables with a dependency graph of maximum degree Δ. That is, whenever two disjoint index sets have no edge between them in the dependency graph, the corresponding collections of random variables are independent. Let $S := \sum_{i=1}^{N} X_i$. Then, for every $t > 0$,*

$$\Pr\left[|S - \mathbb{E}S| \geq t\right] \leq 2\exp\left(-\frac{2t^2}{(\Delta+1)N}\right).$$

Proof. Let χ be the chromatic number of the dependency graph. Since the maximum degree is Δ, we have $\chi \leq \Delta + 1$. Let $I_1, \ldots, I_\chi$ be a proper coloring of the dependency graph and define

$$S_j := \sum_{i \in I_j} X_i.$$

Within each color class I_j, the variables are mutually independent. For $\lambda > 0$, Hölder's inequality gives

$$\mathbb{E}\exp\left(\lambda(S - \mathbb{E}S)\right) = \mathbb{E}\prod_{j=1}^{\chi} \exp\left(\lambda(S_j - \mathbb{E}S_j)\right) \leq \prod_{j=1}^{\chi} \left[\mathbb{E}\exp\left(\chi\lambda(S_j - \mathbb{E}S_j)\right)\right]^{1/\chi}.$$

Since the variables in each S_j are independent and lie in $[0,1]$, Hoeffding's lemma gives

$$\mathbb{E}\exp\left(\chi\lambda(S_j - \mathbb{E}S_j)\right) \le \exp\left(\frac{\chi^2\lambda^2|I_j|}{8}\right).$$

Therefore,

$$\mathbb{E}\exp\left(\lambda(S - \mathbb{E}S)\right) \le \exp\left(\frac{\chi\lambda^2}{8}\sum_{j=1}^{\chi}|I_j|\right) = \exp\left(\frac{\chi\lambda^2 N}{8}\right).$$

Chernoff's bound yields

$$\Pr\left[S - \mathbb{E}S \ge t\right] \le \exp\left(-\frac{2t^2}{\chi N}\right) \le \exp\left(-\frac{2t^2}{(\Delta+1)N}\right).$$

The lower tail is identical. Combining the two tails proves the claim. □

B.5 Profile Homophily of ℓ-Hop Top-k Selection

Theorem B.5 (Profile Homophily of ℓ-Hop Top-k Selection)**.** *Let $G \sim \mathrm{SBM}(n,\pi,B)$ be a C-block stochastic block model with fixed block sizes $n_a = \pi_a n$, where $\pi_a > 0$ and $\sum_{a=1}^{C}\pi_a = 1$. Assume $B \in [0,1]^{C\times C}$ is symmetric. Conditional on the labels, edges are independent and $\Pr[A_{uv} = 1 \mid y_u = a, y_v = b] = B_{ab}$. Let $D_\pi := \mathrm{diag}(\pi_1,\dots,\pi_C)$. Fix an integer $\ell \ge 2$ independent of n. For two distinct nodes u, v, define the raw ℓ-hop reinforcement score $s_\ell(u,v) := (A^\ell)_{uv}$, i.e., the number of length-ℓ walks from u to v. Define the ℓ-hop profile matrix $K_\ell := B(D_\pi B)^{\ell-1}$, and assume that K_ℓ separates same-label pairs from different-label pairs:*

$$\Delta_\ell := \min_{a\in[C]}\left[(K_\ell)_{aa} - \max_{b\ne a}(K_\ell)_{ab}\right] > 0.$$

Then, with probability $1 - o(1)$, for every seed node $u \in V$ and every pair of distinct nodes $v, w \in V \setminus \{u\}$ satisfying $y_v = y_u$ and $y_w \ne y_u$, we have $s_\ell(u,v) > s_\ell(u,w)$. Consequently, for every $k \le \min_{a\in[C]}(n_a - 1)$, the directed top-$k$ graph obtained by connecting each node u to the k nodes $v \ne u$ with the largest raw score $s_\ell(u,v)$ contains only same-label directed edges. Its symmetrization is therefore perfectly homophilic: $h_{G^\star_\ell(k)} = 1$ with probability $1 - o(1)$.

Proof. We first establish a finite-sample concentration bound. The asymptotic statement follows because $\Delta_\ell > 0$ implies the finite-sample margin condition for all sufficiently large n.

Throughout the proof, probabilities and expectations are conditional on the fixed labels; we suppress this conditioning from the notation.

Step 1: ℓ-Hop Profile Mean. Fix two distinct nodes u, v with labels $y_u = a$ and $y_v = b$. The raw score

$$s_\ell(u,v) = (A^\ell)_{uv}$$

counts length-ℓ walks

$$u = x_0, x_1, \dots, x_{\ell-1}, x_\ell = v.$$

First consider walks whose intermediate nodes $x_1,\dots,x_{\ell-1}$ are all distinct and different from u or v. For a label sequence

$$c_1,\dots,c_{\ell-1} \in [C],$$

the number of such intermediate tuples with $y_{x_i} = c_i$ is

$$n^{\ell-1}\pi_{c_1}\cdots\pi_{c_{\ell-1}} + O_\ell(n^{\ell-2}).$$

For each such tuple, conditional independence of SBM edges gives

$$\mathbb{E}\left[A_{ux_1}A_{x_1x_2}\cdots A_{x_{\ell-1}v}\right] = B_{ac_1}B_{c_1c_2}\cdots B_{c_{\ell-1}b}.$$

Therefore the expected number of simple length-ℓ paths from u to v is

$$n^{\ell-1}\sum_{c_1,\dots,c_{\ell-1}}\pi_{c_1}\cdots\pi_{c_{\ell-1}}B_{ac_1}B_{c_1c_2}\cdots B_{c_{\ell-1}b} + O_\ell(n^{\ell-2}).$$

The summation is exactly the (a, b) entry of

$$B(D_\pi B)^{\ell-1} = K_\ell.$$

The remaining length-ℓ walks, namely those with repeated vertices or with an intermediate vertex equal to u or v, are at most $O_\ell(n^{\ell-2})$ in number. Since each contributes at most one, their total contribution to the expectation is also $O_\ell(n^{\ell-2})$. Hence,

$$\mu_{ab}^{(\ell,n)} = n^{\ell-1}(K_\ell)_{ab} + O_\ell(n^{\ell-2}).$$

Step 2: Uniform Concentration of Raw ℓ-Hop Scores. Let $\widetilde{s}_\ell(u, v)$ denote the number of simple length-ℓ paths from u to v. We can write

$$\widetilde{s}_\ell(u, v) = \sum_{\mathbf{x}} Z_{\mathbf{x}},$$

where $\mathbf{x} = (x_1, \ldots, x_{\ell-1})$ ranges over valid simple intermediate tuples and

$$Z_{\mathbf{x}} := A_{ux_1} A_{x_1x_2} \cdots A_{x_{\ell-1}v}.$$

Each $Z_{\mathbf{x}}$ is a Bernoulli random variable.

Construct a dependency graph on the variables $\{Z_{\mathbf{x}}\}$ by connecting two variables whenever their corresponding paths share at least one edge. If two collections of variables have no edge between them in this dependency graph, then their underlying sets of SBM edge variables are disjoint. Hence, the collections are independent, so this is a valid dependency graph.

The number of variables is

$$N = O_\ell(n^{\ell-1}).$$

The maximum degree of the dependency graph satisfies

$$\Delta = O_\ell(n^{\ell-2}).$$

Indeed, fix one simple length-ℓ path from u to v. Another simple length-ℓ path can share one of its ℓ edges. If the shared edge is the first edge incident to u, or the last edge incident to v, then one intermediate coordinate is fixed and the remaining $\ell - 2$ intermediate coordinates are free, giving $O_\ell(n^{\ell-2})$ choices. If the shared edge is internal, then two intermediate coordinates are fixed and only $\ell - 3$ remain free, giving $O_\ell(n^{\ell-3})$ choices. The endpoint-edge cases dominate, so

$$\Delta = O_\ell(n^{\ell-2}).$$

For $\ell = 2$, distinct simple two-hop paths between fixed endpoints share no edge, so the dependency degree is zero; this is also covered by the bound above.

By Lemma B.4, for a sufficiently large constant $C_\ell > 0$ depending only on ℓ and C, for every fixed ordered pair (u, v),

$$\Pr\left[|\widetilde{s}_\ell(u, v) - \mathbb{E}\widetilde{s}_\ell(u, v)| > C_\ell n^{\ell-\frac{3}{2}}\sqrt{\log n}\right] \le 2n^{-4}.$$

Indeed,

$$N(\Delta + 1) = O_\ell(n^{2\ell-3}),$$

so the deviation scale from Lemma B.4 is

$$\sqrt{N(\Delta + 1)\log n} = O_\ell\left(n^{\ell-\frac{3}{2}}\sqrt{\log n}\right).$$

Taking a union bound over at most n^2 ordered pairs, with probability at least $1 - 2n^{-2}$,

$$|\widetilde{s}_\ell(u, v) - \mathbb{E}\widetilde{s}_\ell(u, v)| \le C_\ell n^{\ell-\frac{3}{2}}\sqrt{\log n}$$

simultaneously for all distinct u, v.

It remains to pass from simple paths to all walks. Let

$$s_\ell(u, v) = \widetilde{s}_\ell(u, v) + R_\ell(u, v),$$

where $R_\ell(u,v)$ counts non-simple length-ℓ walks from u to v. As in Step 1,

$$0 \le R_\ell(u,v) \le C'_\ell n^{\ell-2}$$

for a constant $C'_\ell > 0$ depending only on ℓ and C. Therefore,

$$0 \le \mathbb{E}R_\ell(u,v) \le C'_\ell n^{\ell-2}.$$

Hence,

$$\left| s_\ell(u,v) - \mu^{(\ell,n)}_{y_u y_v} \right| \le |\widetilde{s}_\ell(u,v) - \mathbb{E}\widetilde{s}_\ell(u,v)| + 2C'_\ell n^{\ell-2}.$$

After increasing C_ℓ if necessary, we obtain the uniform bound

$$\left| s_\ell(u,v) - \mu^{(\ell,n)}_{y_u y_v} \right| \le C_\ell n^{\ell-\frac{3}{2}} \sqrt{\log n} + C_\ell n^{\ell-2} =: \varepsilon_{n,\ell}$$

simultaneously for all distinct u, v, with probability at least $1 - 2n^{-2}$.

Step 3: Uniform Separation of Same-Label and Different-Label Scores. Define the finite-sample class-separation margin

$$\gamma_{n,\ell} := \min_{a \in [C]} \left[\mu^{(\ell,n)}_{aa} - \max_{b \ne a} \mu^{(\ell,n)}_{ab} \right].$$

By Step 1, $\gamma_{n,\ell} = n^{\ell-1}\Delta_\ell + O_\ell(n^{\ell-2})$. Assume the concentration event from Step 2 holds. Fix a seed node u with label $y_u = a$. Let v satisfy $y_v = a$, and let w satisfy $y_w = b \ne a$. Then

$$s_\ell(u,v) \ge \mu^{(\ell,n)}_{aa} - \varepsilon_{n,\ell},$$

while

$$s_\ell(u,w) \le \mu^{(\ell,n)}_{ab} + \varepsilon_{n,\ell} \le \mu^{(\ell,n)}_{aa} - \gamma_{n,\ell} + \varepsilon_{n,\ell}.$$

Since $\Delta_\ell > 0$, we have $\gamma_{n,\ell} = n^{\ell-1}\Delta_\ell + O_\ell(n^{\ell-2})$, while $\varepsilon_{n,\ell} = O_\ell(n^{\ell-3/2}\sqrt{\log n})$. Hence, $\gamma_{n,\ell} > 2\varepsilon_{n,\ell}$ for all sufficiently large n and

$$s_\ell(u,v) > s_\ell(u,w).$$

In particular, $s_\ell(u,v) > s_\ell(u,w)$ holds whenever the finite-sample margin satisfies $\gamma_{n,\ell} > 2\varepsilon_{n,\ell}$, with $\varepsilon_{n,\ell} = C_\ell n^{\ell-3/2}\sqrt{\log n} + C_\ell n^{\ell-2}$. Thus, uniformly over every seed node, every same-label candidate has strictly larger raw ℓ-hop score than every different-label candidate.

Step 4: Homophily of the Top-k Graph. Assume

$$k \le \min_{a \in [C]} (n_a - 1).$$

Then every seed node u has at least k other nodes with the same label. By Step 3, all same-label candidates have larger score $s_\ell(u,\cdot)$ than all different-label candidates. Therefore, every directed top-k edge $u \to v$ satisfies

$$y_u = y_v.$$

Symmetrization cannot create a cross-label edge, since an undirected edge $\{u,v\}$ is included only if at least one of the directed edges $u \to v$ or $v \to u$ was selected. Hence, the symmetrized graph is perfectly homophilic:

$$h_{G^\star_\ell(k)} = 1.$$

Finally, suppose $\Delta_\ell > 0$. By Step 1,

$$\mu^{(\ell,n)}_{ab} = n^{\ell-1}(K_\ell)_{ab} + O_\ell(n^{\ell-2}).$$

Therefore,

$$\gamma_{n,\ell} = n^{\ell-1}\Delta_\ell + O_\ell(n^{\ell-2}).$$

Since

$$n^{\ell-1} \gg n^{\ell-\frac{3}{2}}\sqrt{\log n} + n^{\ell-2},$$

$\gamma_{n,\ell} > 2\varepsilon_{n,\ell}$ holds for all sufficiently large n. Hence, the perfect-homophily conclusion holds with probability $1 - o(1)$. □

B.6 Proof of Lemma 4.9 (Effective Resistance Characterization)

Activation Times and the Activation DAG. For every $x \in A^v$, define its activation time

$$T(x) := \min\{t \in \{0, 1, \ldots, \ell\} : x \in S_t^v\}.$$

Thus, $T(x) = 0$ for $x \in S_0^v$, and $T(x) \geq 1$ for every non-initially active vertex. Construct a directed graph $D^v = (A^v, \vec{E}^{\,v})$ as follows. For every edge $\{a, b\} \in E(G_v)$ with $T(a) < T(b)$, include the directed arc $a \to b$. If $T(a) = T(b)$, include no arc corresponding to $\{a, b\}$. Thus, every arc strictly increases activation time, so D^v is acyclic.

Cut Lower Bound on Subsets Disjoint From S_0^v. We first prove a cut lower bound. Let $U \subseteq A^v$ be nonempty and suppose $U \cap S_0^v = \emptyset$. Choose $x \in U$ minimizing $T(x)$ over all vertices in U. Since $U \cap S_0^v = \emptyset$, we have $T(x) \geq 1$. By minimality of $T(x)$, no vertex of U is active before time $T(x)$, so

$$U \cap S_{T(x)-1}^v = \emptyset.$$

Because x first activates at time $T(x)$, the uniform threshold condition gives

$$\left|N(x) \cap S_{T(x)-1}^v\right| \geq \kappa.$$

Every vertex $y \in N(x) \cap S_{T(x)-1}^v$ lies outside U, and satisfies $T(y) < T(x)$. Hence, the edge $\{y, x\}$ appears in D^v as the incoming arc $y \to x$. Therefore, at least κ arcs enter U:

$$\left|\delta_{D^v}^-(U)\right| \geq \left|N(x) \cap S_{T(x)-1}^v\right| \geq \kappa,$$

where

$$\delta_{D^v}^-(U) := \{(a, b) \in \vec{E}^{\,v} : a \in A^v \setminus U,\ b \in U\}.$$

Augmented Network and Capacity Assignment. We now add a new source vertex s. For every $z \in S_0^v$, add an arc $s \to z$ of capacity κ, and give every arc of D^v capacity 1. We claim that every s-u cut in this network has capacity at least κ.

Lower Bound on Every s-u Cut. Let X be the source side of an arbitrary s-u cut, so $s \in X$ and $u \notin X$. Define

$$U := A^v \setminus X.$$

Then $u \in U$, so $U \neq \emptyset$. If $U \cap S_0^v \neq \emptyset$, then for any $z \in U \cap S_0^v$, the source arc $s \to z$ crosses the cut and has capacity κ. Hence, the cut has capacity at least κ.

It remains to consider the case $U \cap S_0^v = \emptyset$. By the cut lower bound proved above,

$$\left|\delta_{D^v}^-(U)\right| \geq \kappa.$$

Each arc in $\delta_{D^v}^-(U)$ crosses from $X \cap A^v$ into $U = A^v \setminus X$, and each such arc has capacity 1. Thus, the cut again has capacity at least κ.

Max-Flow/Min-Cut and Integral Decomposition. Therefore, every s-u cut has capacity at least κ. By the max-flow/min-cut theorem, the maximum s-u flow has value at least κ. Since all capacities are integral, there is an integral flow of value at least κ. Decomposing this flow into directed s-u paths and selecting any κ unit-flow paths, we obtain κ directed paths from s to u. The source arcs $s \to z$ may be used more than once, but every arc of D^v has unit capacity, so the portions of these paths inside D^v are pairwise arc-disjoint.

Removing the initial source arcs gives κ pairwise arc-disjoint directed paths in D^v, each starting at some vertex of S_0^v and ending at u. Since D^v contains at most one orientation of any undirected edge of G_v, these directed paths correspond to κ pairwise edge-disjoint undirected paths in G_v.

Length Bound. Finally, along every directed path in D^v, activation time strictly increases at each arc. A directed path starting in S_0^v starts at activation time 0 and ends at activation time $T(u)$. Hence, it has at most $T(u)$ arcs, and so each resulting undirected path has length at most $T(u) \leq \ell$.

B.7 Proof of Lemma 4.10

If some path P_i intersects S in more than one vertex, replace P_i by the suffix of P_i beginning at its last vertex in S and ending at u. This operation does not increase its length and preserves pairwise edge-disjointness. Hence, we may assume that each P_i meets S only at its initial endpoint.

Orient each path P_i from u toward its endpoint in S, and send $1/\kappa$ units of current along P_i. Since the paths are pairwise edge-disjoint, every used edge carries current of magnitude exactly $1/\kappa$, and all unused edges carry zero current. The superposition of these path flows is a unit flow from u to the set S: the total outflow from u is 1, the total inflow into S is 1, and flow is conserved at every vertex in $W \setminus (S \cup \{u\})$.

By Thomson's principle, effective resistance is the minimum energy over all unit flows from u to S. Therefore,

$$R_{\text{eff}}^H(S,u) \le \sum_{e\in F} f(e)^2 = \sum_{i=1}^{\kappa} \ell_i \left(\frac{1}{\kappa}\right)^2 = \frac{1}{\kappa^2}\sum_{i=1}^{\kappa} \ell_i.$$

If $\ell_i \le L$ for all i, this gives

$$R_{\text{eff}}^H(S,u) \le \frac{1}{\kappa^2}\cdot \kappa L = \frac{L}{\kappa}.$$

B.8 Proof of Theorem 4.8

Fix $u \in S_\ell^v \setminus S_0^v$. By Lemma 4.9, the induced activated subgraph G_v contains κ pairwise edge-disjoint paths from S_0^v to u, each of length at most $T(u)$. Applying Lemma 4.10 with $H = G_v$, $S = S_0^v$, and $L = T(u)$, we obtain

$$R_{\text{eff}}^{G_v}(S_0^v, u) \le \frac{T(u)}{\kappa}.$$

Since u activates by time ℓ, $T(u) \le \ell$. Hence,

$$R_{\text{eff}}^{G_v}(S_0^v, u) \le \frac{T(u)}{\kappa} \le \frac{\ell}{\kappa}.$$

B.9 Finite-Sample Cascade Scores and Short-Walk Reinforcement

In the main text we define the short-walk event $\mathcal{F}_L := \{(A^2+\cdots+A^L)_{uv} \ge \kappa\}$ as a population-level proxy for the empirical co-activation score $f_v(u)$. The following result makes this relation precise. It is stated conditional on the input graph G; if G is random, the same statement holds conditionally on G.

Theorem B.6 (Empirical Cascade Scores and Short-Walk Support)**.** *Let $G = (V, E)$ be a fixed undirected graph with adjacency matrix A and $|V| = n$. For each seed $v \in V$ with $N_v \ge 1$, let $S_{v,b}$ denote the final active set of cascade realization b, and define $\widehat{p}_v(u) := \frac{1}{N_v}\sum_{b=1}^{N_v} \mathbf{1}\{u \in S_{v,b} \setminus \{v\}\}$, $p_v(u) := \mathbb{E}\big[\widehat{p}_v(u)\big]$. Assume that, for each such seed v, the N_v realizations can be partitioned into independent blocks $B_{v,1}, \dots, B_{v,m_v}$ with sizes $a_{v,j} = |B_{v,j}|$; dependence within a block is permitted. Let $N_{\text{eff}}(v) := \frac{\left(\sum_{j=1}^{m_v} a_{v,j}\right)^2}{\sum_{j=1}^{m_v} a_{v,j}^2} = \frac{N_v^2}{\sum_{j=1}^{m_v} a_{v,j}^2}$ denote the effective number of independent cascade samples for seed v. Then the following statements hold.*

(i) (Concentration.) *For any $\delta \in (0,1)$, with probability at least $1-\delta$, for every ordered pair (v,u) with $u \ne v$ and $N_v \ge 1$, $\big|\widehat{p}_v(u) - p_v(u)\big| \le \varepsilon_v(\delta) := \sqrt{\frac{\log\left(2n(n-1)/\delta\right)}{2N_{\text{eff}}(v)}}$. Equivalently, $|f_v(u) - N_v p_v(u)| \le N_v \varepsilon_v(\delta)$ uniformly over all such pairs.*

(ii) (Top-k Stability.) *Fix a seed v and an integer k with $1 \le k \le n-2$. Let $T_k(v)$ denote the population top-k set under p_v and $\widehat{T}_k(v)$ the empirical top-k set under $\widehat{p}_v$, using a common deterministic tie-breaking rule. Let $p_{v,(1)} \ge p_{v,(2)} \ge \cdots$ be the sorted values of $\{p_v(u) : u \ne v\}$, and define the population margin $\gamma_v := p_{v,(k)} - p_{v,(k+1)}$. If $\gamma_v > 2\varepsilon_v(\delta)$, then $\widehat{T}_k(v) = T_k(v)$ on the event of part* (i)*.*

(iii) (Short-Walk Certificate for Fixed-Threshold TAS.) *Consider fixed-threshold TAS with $\tau \ge 1$, initialization $S_0^v = \{v\} \cup R$ for some $R \subseteq N(v)$, and at most ℓ post-initialization vertex activations. The conclusion below applies both to the clean sequential TAS rule of Section 2.1 and to the hub-restricted implementation, provided that any activation requires at least τ distinct already-active support vertices. Then for every nonlocal node $u \notin N[v]$, $f_v(u) > 0$ implies $\sum_{s=2}^{\ell+1}(A^s)_{vu} \ge \tau$.*

The same implication holds with $p_v(u) > 0$ in place of $f_v(u) > 0$. When counts are aggregated over a threshold set $\mathcal{T}$, the implication holds unconditionally with $\tau_{\min} := \min \mathcal{T}$; if threshold-specific counts $f_v^{(\tau)}(u)$ are retained, the implication holds for that threshold τ.

Proof. For *(i)*, fix an ordered pair (v, u), $u \neq v$, with $N_v \geq 1$. For block j, define

$$Y_{v,j}(u) := \sum_{b \in B_{v,j}} \mathbf{1}\{u \in S_{v,b} \setminus \{v\}\}.$$

The variables $Y_{v,1}(u), \ldots, Y_{v,m_v}(u)$ are independent by assumption and satisfy $0 \leq Y_{v,j}(u) \leq a_{v,j}$. Since

$$\widehat{p}_v(u) = \frac{1}{N_v} \sum_{j=1}^{m_v} Y_{v,j}(u),$$

Hoeffding's inequality for independent bounded random variables gives, for every $\epsilon > 0$,

$$\Pr\left[\left|\widehat{p}_v(u) - p_v(u)\right| \geq \epsilon\right] \leq 2\exp\left(-\frac{2N_v^2\epsilon^2}{\sum_{j=1}^{m_v} a_{v,j}^2}\right) = 2\exp\left(-2N_{\text{eff}}(v)\epsilon^2\right).$$

Taking

$$\epsilon = \sqrt{\frac{\log\left(2n(n-1)/\delta\right)}{2N_{\text{eff}}(v)}}$$

makes the failure probability at most $\delta/(n(n-1))$. A union bound over all $n(n-1)$ ordered pairs (v, u), $u \neq v$, proves *(i)*. Pairs with seeds for which no cascades are run are excluded by the condition $N_v \geq 1$.

For *(ii)*, work on the event from *(i)* and fix v. For any $a \in T_k(v)$ and $b \notin T_k(v)$, the population margin gives $p_v(a) - p_v(b) \geq \gamma_v$. Therefore,

$$\widehat{p}_v(a) - \widehat{p}_v(b) \geq p_v(a) - p_v(b) - 2\varepsilon_v(\delta) \geq \gamma_v - 2\varepsilon_v(\delta) > 0.$$

Thus, every population top-k node has a strictly larger empirical score than every non-top-k node, so the empirical and population top-k sets coincide.

For *(iii)*, consider a single TAS-τ realization from seed v. Order the post-initialization activations by their insertion time. We first show by induction that, after r post-initialization activations, every active vertex other than v itself has a walk from v of length at most $r + 1$, while v itself has a trivial length-zero walk. At initialization, every vertex in $R \subseteq N(v)$ has a length-one walk from v, so the claim holds for $r = 0$. If x is the $(r+1)$-st activated vertex, the TAS rule activates x only after at least one already-active support vertex $z \in N(x)$ has been observed, because $\tau \geq 1$. By the induction hypothesis, z has a walk from v of length at most $r + 1$, or $z = v$ has a length-zero walk. Appending the edge (z, x) gives a walk from v to x of length at most $r + 2$. This proves the induction claim.

Now let $u \notin N[v]$ be activated as the r-th post-initialization activation, where $1 \leq r \leq \ell$. Since $u \notin N[v]$, it was not active at initialization. At the activation time of u, the rule has observed at least τ distinct already-active support vertices $z_1, \ldots, z_\tau \in N(u)$. By the induction claim, each z_i has a walk from v of length at most r, or a length-zero walk if $z_i = v$. The latter case is impossible because $z_i \in N(u)$ and $u \notin N[v]$ imply $z_i \neq v$. Appending the distinct final edges (z_i, u) gives τ walks from v to u, each of length at most $r + 1 \leq \ell + 1$. These walks are distinct because they end with distinct second-to-last vertices z_i. Because $u \notin N[v]$, each walk has length at least two. Hence,

$$\sum_{s=2}^{\ell+1} (A^s)_{vu} \geq \tau.$$

If $f_v(u) > 0$, some sampled realization activated u, so the deterministic argument above applies. If $p_v(u) > 0$, then a realization activating u occurs with positive probability, and the same certificate follows. Under the nonzero-count convention of (1), every selected nonlocal directed edge $v \to u$ has $f_v(u) > 0$, so the certificate applies to every such edge of G^*. □

Corollary B.7 (TAS Lower Bound From Two-Hop Common Neighbors)**.** *Consider TAS with threshold τ, and suppose the initial active set is $S_0^v = \{v\} \cup R$, where R is drawn uniformly from the r_v-subsets of $N(v)$ with $r_v \le d(v)$. For a nonlocal node $u \notin N[v]$, define the number of two-hop common neighbors*

$$h_{vu} := |N(v) \cap N(u)| = (A^2)_{vu},$$

and the hypergeometric tail

$$\Phi_{\tau,r_v,d(v)}(h) := \sum_{i=\tau}^{\min\{r_v,h\}} \frac{\binom{h}{i}\binom{d(v)-h}{r_v-i}}{\binom{d(v)}{r_v}}.$$

The function $h \mapsto \Phi_{\tau,r_v,d(v)}(h)$ is nondecreasing.

(i) (Parallel Activation.) *If TAS activates all initially eligible vertices in parallel, then for every nonlocal $u \notin N[v]$,*

$$p_v(u) \ge \Phi_{\tau,r_v,d(v)}\big((A^2)_{vu}\big).$$

(ii) (Sequential Uniform Activation.) *Suppose instead that, at each step, one initially eligible vertex is selected uniformly at random. Let*

$$M_v := \max\Big\{1, \max_R \big|\{x \in V \setminus S_0^v : \sigma_0(x) \ge \tau\}\big|\Big\}$$

denote the maximum size of the initially eligible set over all initializations R, taken to be at least one. Then for every nonlocal $u \notin N[v]$,

$$p_v(u) \ge \frac{1}{M_v}\,\Phi_{\tau,r_v,d(v)}\big((A^2)_{vu}\big).$$

On the concentration event of Theorem B.6, the same lower bounds hold for the empirical score $\widehat{p}_v(u)$ up to a subtractive correction of $\varepsilon_v(\delta)$.

Proof. Let S_v denote the final active set of a generic clean-TAS realization from seed v. For $u \notin N[v]$, the initial support of u is

$$\sigma_0(u) = |N(u) \cap R|.$$

Among the $d(v)$ vertices in $N(v)$, exactly

$$h_{vu} = |N(v) \cap N(u)| = (A^2)_{vu}$$

are also neighbors of u. Since R is sampled uniformly among all r_v-subsets of $N(v)$,

$$\sigma_0(u) \sim \mathrm{Hypergeom}\big(d(v), h_{vu}, r_v\big).$$

Therefore,

$$\Pr\left[\sigma_0(u) \ge \tau\right] = \Phi_{\tau,r_v,d(v)}(h_{vu}).$$

Parallel Activation. Assume first that clean TAS activates all initially eligible vertices in parallel, and that at least one activation layer is performed. On the event $\{\sigma_0(u) \ge \tau\}$, the vertex u is eligible at the first layer. Hence, u activates at the first layer, so

$$\{\sigma_0(u) \ge \tau\} \subseteq \{u \in S_v\}.$$

Since $u \ne v$, this gives

$$p_v(u) = \Pr\left[u \in S_v \setminus \{v\}\right] \ge \Pr\left[\sigma_0(u) \ge \tau\right] = \Phi_{\tau,r_v,d(v)}(h_{vu}).$$

Sequential Uniform Activation. Now suppose instead that clean TAS activates one initially eligible vertex at a time, choosing uniformly among the initially eligible vertices, and that at least one activation step is performed. For an initialization R, let

$$\mathcal{E}(R) := \{x \in V \setminus S_0^v : \sigma_0(x) \ge \tau\}$$

be the initially eligible set. Define

$$M_v := \max\left\{1, \max_R |\mathcal{E}(R)|\right\}.$$

Condition on the event $\{\sigma_0(u) \geq \tau\}$. Since $u \notin N[v]$, we have $u \notin S_0^v$, and so $u \in \mathcal{E}(R)$. Thus,

$$1 \leq |\mathcal{E}(R)| \leq M_v.$$

Because the first activated vertex is chosen uniformly from $\mathcal{E}(R)$, the conditional probability that u is selected at the first activation step is

$$\frac{1}{|\mathcal{E}(R)|} \geq \frac{1}{M_v}.$$

Therefore,

$$\begin{aligned} p_v(u) &\geq \Pr\left[u \text{ is selected at the first activation step}\right] \\ &= \mathbb{E}\left[\mathbf{1}\{\sigma_0(u) \geq \tau\}\frac{1}{|\mathcal{E}(R)|}\right] \geq \frac{1}{M_v}\Pr\left[\sigma_0(u) \geq \tau\right]. \end{aligned}$$

Using the hypergeometric identity above,

$$p_v(u) \geq \frac{1}{M_v}\Phi_{\tau,r_v,d(v)}(h_{vu}).$$

Monotonicity of Φ. To prove monotonicity, couple the hypergeometric experiment with h marked elements to the one with $h+1$ marked elements by changing one unmarked element into a marked element and using the same sampled subset. Under this coupling, the number of marked sampled elements cannot decrease, so the probability of sampling at least τ marked elements is nondecreasing in h. Therefore, $\Phi_{\tau,r_v,d(v)}(h)$ is nondecreasing in h.

Empirical Bounds. On the concentration event of Theorem B.6,

$$|\widehat{p}_v(u) - p_v(u)| \leq \varepsilon_v(\delta)$$

simultaneously over all ordered pairs (v,u) with $u \neq v$ and $N_v \geq 1$. Therefore, for parallel clean TAS,

$$\widehat{p}_v(u) \geq p_v(u) - \varepsilon_v(\delta) \geq \Phi_{\tau,r_v,d(v)}(h_{vu}) - \varepsilon_v(\delta),$$

and for sequential uniform clean TAS,

$$\widehat{p}_v(u) \geq p_v(u) - \varepsilon_v(\delta) \geq \frac{1}{M_v}\Phi_{\tau,r_v,d(v)}(h_{vu}) - \varepsilon_v(\delta).$$

Since the concentration event is uniform, these empirical bounds hold simultaneously for all nonlocal pairs $u \notin N[v]$ and all seeds v with $N_v \geq 1$, with probability at least $1-\delta$. □

One-round, parallel, nonlocal TAS. Fix integers $1 \leq \tau \leq r$, independent of n. For each seed v, let $U_v := V \setminus N[v]$ and $r_v := \min\{r, d(v)\}$. Draw $N_v \geq 1$ subsets $R_{v,b}$, each uniform among the r_v-subsets of $N(v)$ conditional on G. In realization b, activate exactly those $u \in U_v$ satisfying $|N(u) \cap R_{v,b}| \geq \tau$, with no further propagation. Let $\widehat{p}_v(u)$ be the empirical activation frequency. Select the k highest positive scores (all positive scores if fewer exist), with fixed deterministic ties. Symmetrize the selected pairs to obtain the unweighted graph $\widehat{G}$; retain no original edges. For each seed, assume independent blocks of realizations conditional on G, with block sizes $a_{v,j}$ and sample counts fixed given G, and set

$$N_{\text{eff}}(v) := N_v^2 / \sum_j a_{v,j}^2.$$

Within-block dependence is allowed; independence across seeds is not required. Independent realizations give $N_{\text{eff}}(v) = N_v$.

Corollary B.8 (End-to-end guarantee for one-round TAS)**.** *Under the fixed-parameter SBM assumptions of Theorem 4.7, including $\Delta_2 > 0$, assume $B_{aa} < 1$ for every class a. For the variant above, there exist constants $c_0, c_1 > 0$, depending only on (B, π, r, τ), such that, for all sufficiently large n, every $\delta \in (0,1)$ and integer $1 \leq k \leq c_0 n$, if*

$$\min_v N_{\text{eff}}(v) \geq c_1 \log(n/\delta),$$

then $\widehat{G}$ is perfectly homophilic with minimum degree at least k, with probability at least $1 - 2n^{-2} - \delta$.

If, conditional on the labels, the features $X_i \sim \mathcal{N}(\mu_{y_i}, \sigma^2 I_d)$ are independent across vertices and jointly independent of the graph and cascade sampling, with $\sigma > 0$ and $\Delta := \min_{a \neq b} \|\mu_a - \mu_b\|_2 > 0$, then classifying the uniform feature average over each vertex and its neighbors in $\widehat{G}$ by the nearest true class mean gives

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\widehat{y}_i \neq y_i\}\right] \leq 2n^{-2} + \delta + (C-1)\exp\left(-\frac{(k+1)\Delta^2}{8\sigma^2}\right).$$

Probability and expectation are conditional on the labels and include all construction and feature randomness.

Proof. We condition throughout on the labels. Write $K := B\,\mathrm{diag}(\pi)B^\top$ and define

$$\overline{d}_a := \sum_b \pi_b B_{ab}, \qquad \theta_{ab} := K_{ab}/\overline{d}_a, \qquad H_{r,\tau}(x) := \sum_{j=\tau}^{r} \binom{r}{j} x^j (1-x)^{r-j},$$

with continuous endpoint values. Let

$$\eta_0 := \min_a \left[H_{r,\tau}(\theta_{aa}) - \max_{b \neq a} H_{r,\tau}(\theta_{ab})\right], \qquad \rho := \min_a \pi_a(1 - B_{aa}).$$

We will take $c_0 := \rho/2$ and $c_1 := 96/\eta_0^2$. First, $0 \leq K_{ab} \leq \overline{d}_a$, since $B_{bc} \in [0,1]$. The positive profile margin implies $K_{aa} > 0$ and hence $\overline{d}_a > 0$. For $1 \leq \tau \leq r$, $H_{r,\tau}$ is strictly increasing on $[0,1]$: on $(0,1)$ its derivative is

$$H'_{r,\tau}(x) = r\binom{r-1}{\tau-1} x^{\tau-1}(1-x)^{r-\tau} > 0.$$

Thus $\theta_{aa} > \theta_{ab}$ for $b \neq a$ implies $\eta_0 > 0$. The condition $B_{aa} < 1$ gives $\rho > 0$.

Exact activation probabilities. For a fixed graph and $u \in U_v$, put $h_{vu} := |N(v) \cap N(u)| = (A^2)_{vu}$. Since $u \notin N[v]$, its initial support comes only from $R_{v,b}$. Consequently, the one-round activation probability is exactly

$$p_v(u) := \mathbb{E}[\widehat{p}_v(u) \mid G] = \sum_{j=\tau}^{\min\{r_v, h_{vu}\}} \frac{\binom{h_{vu}}{j}\binom{d(v)-h_{vu}}{r_v-j}}{\binom{d(v)}{r_v}}.$$

We use the conventions that an empty sum is zero and that a binomial coefficient with an invalid lower argument is zero. This is an equality because only the initial parallel round is used.

Uniform SBM event. For $y_v = a$ and $y_u = b$, $u \neq v$, the conditional expectations satisfy

$$\mathbb{E}[d(v)] = n\overline{d}_a - B_{aa}, \qquad \mathbb{E}[h_{vu}] = nK_{ab} - B_{ab}(B_{aa} + B_{bb}).$$

The number $M_v := |\{u \in U_v : y_u = y_v\}|$ of same-class nonlocal candidates has expectation $(n_a - 1)(1 - B_{aa})$. Each of $d(v)$, h_{vu}, and M_v is a sum of at most n independent Bernoulli variables conditional on the labels. For h_{vu}, the summands are $A_{vw}A_{wu}$ over distinct intermediate vertices w; these summands use disjoint edges. Hoeffding's inequality with $t_n := \sqrt{3n\log n}$ and a union bound over all these variables gives a simultaneous event $\mathcal{E}_n$ with

$$\Pr[(]\,\mathcal{E}_n^c) \leq 2n^{-4} + 4n^{-5} \leq 2n^{-2} \qquad (n \geq 2),$$

on which all deviations from their expectations are at most t_n. Therefore, uniformly over vertices and distinct pairs,

$$\frac{d(v)}{n} = \overline{d}_a + O\left(\sqrt{\frac{\log n}{n}}\right), \qquad \frac{h_{vu}}{n} = K_{ab} + O\left(\sqrt{\frac{\log n}{n}}\right),$$

and, for all sufficiently large n, $d(v) \geq r$ and $M_v \geq \rho n/2 \geq k$.

For fixed r and $d(v) \geq r$, sampling with and without replacement can be coupled until the first repeated draw. Hence

$$\left|p_v(u) - H_{r,\tau}\left(\frac{h_{vu}}{d(v)}\right)\right| \leq \frac{r(r-1)}{2d(v)}.$$

Since $\min_a \overline{d}_a > 0$ and $H_{r,\tau}$ is a fixed polynomial, it follows that, on $\mathcal{E}_n$ and for all sufficiently large n,

$$\sup_{v \in V,\ u \in U_v} |p_v(u) - H_{r,\tau}(\theta_{y_v y_u})| \le \frac{\eta_0}{4}.$$

Thus every same-class nonlocal candidate has population score at least $3\eta_0/4$, and every same-class nonlocal score exceeds every cross-class nonlocal score for the same seed by at least $\eta_0/2$. The latter comparison is vacuous when no cross-class candidate exists.

Finite-sample selection. For $n \ge 2$ and $\delta \in (0,1)$, $\log(2n(n-1)/\delta) \le 3\log(n/\delta)$, so our sampling condition implies $N_{\text{eff}}(v) \ge 32\eta_0^{-2}\log(2n(n-1)/\delta)$ for every seed. Conditional on G, the sum of activation indicators within block j lies in $[0, a_{v,j}]$, and these block sums are independent for a fixed seed. Hoeffding's inequality followed by a union bound over at most $n(n-1)$ ordered pairs yields, with conditional probability at least $1-\delta$,

$$|\widehat{p}_v(u) - p_v(u)| \le \sqrt{\frac{\log(2n(n-1)/\delta)}{2N_{\text{eff}}(v)}} \le \frac{\eta_0}{8} \quad \text{for every } v \in V,\ u \in U_v.$$

On this event and $\mathcal{E}_n$, every same-class nonlocal score is strictly positive, and every such score exceeds every cross-class nonlocal score by at least $\eta_0/4$. Since $M_v \ge k$, each seed selects exactly k same-class vertices. Symmetrization preserves label agreement and cannot remove these k neighbors. The combined construction-failure probability is at most $2n^{-2} + \delta$.

Classification. Let $q_i := 1 + |N_{\widehat{G}}(i)|$ and $Z_i := q_i^{-1} \sum_{j \in \{i\} \cup N_{\widehat{G}}(i)} X_j$. The classifier is $\widehat{y}_i := \arg\min_a \|Z_i - \mu_a\|_2^2$, with fixed deterministic ties. Condition on a successful output graph and the labels. The construction uses the original graph and feature-independent sampling randomness, so the selected features remain independent Gaussians conditional on that output. Every selected neighbor of i has label y_i and $q_i \ge k+1$, giving

$$Z_i \sim \mathcal{N}\left(\mu_{y_i}, \frac{\sigma^2}{q_i} I_d\right).$$

For an incorrect class $b \ne y_i$, the nearest-mean decision can prefer b only if a one-dimensional Gaussian projection crosses half the squared mean separation. The standard Gaussian tail bound gives

$$\Pr\left[\|Z_i - \mu_b\|_2^2 \le \|Z_i - \mu_{y_i}\|_2^2 \mid \widehat{G},\ \text{successful construction}, y\right] \le \exp\left(-\frac{q_i \|\mu_{y_i} - \mu_b\|_2^2}{8\sigma^2}\right).$$

A union bound over incorrect classes gives the stated single-node tail. Averaging over nodes and bounding the loss by one on construction failure proves the expected-error bound. The construction-failure term is included only once. □

B.10 Edge-Connectivity Certificate for Fixed-Threshold TAS

Theorem 4.8 bounds the effective resistance of threshold activations. The same activation-path argument yields a complementary edge-connectivity consequence, which we record here. Throughout this section, κ denotes a single fixed threshold; when the final TAS graph aggregates over multiple thresholds, the certificate below applies to each cascade run separately, with κ taken to be the threshold of the contributing run.

Definition B.9 (Set-To-Vertex Edge Connectivity)**.** *Let $G = (V, E)$ be an undirected graph. For a nonempty set $S \subseteq V$ and a vertex $u \in V \setminus S$, define*

$$\lambda_G(S, u) := \min\{|F| : F \subseteq E,\ \textit{no path from } S \textit{ to } u \textit{ in } G \setminus F\}.$$

By Menger's theorem, $\lambda_G(S,u)$ equals the maximum number of pairwise edge-disjoint paths from S to u in G.

Let $F_k^\kappa(v)$ denote the directed top-k cascade-neighbor set obtained from positive co-activation counts in TAS-κ runs seeded at v. A selected directed edge $v \to u$ is *nonlocal* if $u \notin N[v]$.

Proposition B.10 (Edge-Connectivity Certificate)**.** *Consider a TAS-κ cascade from seed v with initial active set $S_0^v = \{v\} \cup R$, where $R \subseteq N(v)$, and walk length ℓ. If a node $u \notin S_0^v$ is activated within ℓ steps, then G contains κ pairwise edge-disjoint paths from S_0^v to u, each of length at most the activation time $T(u) \leq \ell$. In particular, $\lambda_G(S_0^v, u) \geq \kappa$. If $v \to u$ is a nonlocal directed edge in the TAS-κ top-k graph, then $\lambda_G(N[v], u) \geq \kappa$.*

For the symmetrized graph, the certificate holds in at least one of the two selected orientations: if $\{v, u\}$ appears because $v \to u$ was selected, then $\lambda_G(N[v], u) \geq \kappa$; if it appears because $u \to v$ was selected, then $\lambda_G(N[u], v) \geq \kappa$.

Proof. Let $u \notin S_0^v$ activate within ℓ steps. By Lemma 4.9, the activated subgraph $G[S_\ell^v]$ contains κ pairwise edge-disjoint paths from S_0^v to u, each of length at most $T(u) \leq \ell$. These paths exist in G as well, so by Menger's theorem $\lambda_G(S_0^v, u) \geq \kappa$.

Suppose now that $v \to u$ is a nonlocal directed edge selected by the TAS-κ top-k rule. Top-k selection requires positive co-activation, so some cascade run from v activated u. Since $S_0^v \subseteq N[v]$ and $u \notin N[v]$, the activation cannot be due to initialization; hence, u was reached by propagation. The first part gives $\lambda_G(S_0^v, u) \geq \kappa$ for that run.

Edge connectivity is monotone in the source set: if $S \subseteq S'$ and $u \notin S'$, then $\lambda_G(S', u) \geq \lambda_G(S, u)$, because every cut separating S' from u also separates S from u. Since $S_0^v \subseteq N[v]$ and $u \notin N[v]$, we obtain $\lambda_G(N[v], u) \geq \lambda_G(S_0^v, u) \geq \kappa$. The symmetrized statement follows by applying the directed argument to whichever orientation was selected before symmetrization. □

Corollary B.11 (Cut Obstruction)**.** *Let $V = A \sqcup B$ be a partition of the vertices and let $E_G(A, B) := \{(a, b) \in E : a \in A,\ b \in B\}$. If $|E_G(A, B)| < \kappa$, $N[v] \subseteq A$, and $u \in B$, then the nonlocal directed edge $v \to u$ cannot be selected by TAS-κ cascade rewiring.*

Proof. The cross-cut $E_G(A, B)$ separates $N[v]$ from u, so $\lambda_G(N[v], u) \leq |E_G(A, B)| < \kappa$. By Proposition B.10, any selected nonlocal directed edge $v \to u$ must satisfy $\lambda_G(N[v], u) \geq \kappa$, a contradiction. □

Corollary B.12 (Reachable-Certificate Set)**.** *For each vertex v, define*

$$B_\kappa(v) := \{u \in V \setminus N[v] : \lambda_G(N[v], u) \geq \kappa\}.$$

Every nonlocal directed neighbor selected by TAS-κ lies in $B_\kappa(v)$:

$$F_k^\kappa(v) \setminus N[v] \subseteq B_\kappa(v).$$

For an undirected edge $\{v, u\}$ in the symmetrized nonlocal graph, at least one of $u \in B_\kappa(v)$ or $v \in B_\kappa(u)$ holds.

Proof. The directed statement is Proposition B.10. The undirected statement follows because $\{v, u\}$ is included only if at least one of $v \to u$ or $u \to v$ was selected. □

Remark B.13. *Proposition B.10 shows that TAS-κ acts as a redundancy filter: it promotes a nonlocal pair only when the target is already connected to the seed's closed neighborhood by at least κ edge-disjoint paths in G. Cascade rewiring therefore reinforces mesoscopic regions with redundant short-path support, but cannot create shortcuts across genuine bottlenecks. This is the structural reason cascade rewiring is complementary to bottleneck-targeted methods such as* FoSR *and* SDRF*, which are designed to add edges across small cuts.*

The certificate is threshold-specific. If counts are aggregated across multiple thresholds, an edge's certificate corresponds to the threshold(s) at which it co-activated; without threshold-specific counts, only the minimum threshold used gives an unconditional certificate. MAS admits an analogous certificate, but only along cascade prefixes where every activation up to the target has support at least κ.

C Constructing the Multiset $\mathcal{M}_v$: Pseudocode and Complexity Analysis

C.1 Pseudocode and Implementation

In this section, we present pseudocode for MAS and TAS procedures introduced in Section 3. These procedures, originally formulated as adjacency-search rules by Chaitanya et al. [2025], are reinterpreted here as *discrete-time* contagion dynamics. MAS and TAS systematically explore node neighborhoods to construct multisets $\mathcal{M}_v$, which capture local and mesoscopic structural information and form the basis for our graph rewiring strategies.

MAS first explores densely connected regions by activating nodes with maximal neighbor support and gradually lowers the activation threshold to reach sparser regions. The procedure relies on the following parameters:

- **Walk Length (ℓ):** Number of steps in the contagion; default $= 10$.
- **Number of Neighbors (r):** Size of the random subset $R \subseteq N(v)$ selected for initialization; default $= 5$.
- **Number of Permutations (P):** Number of times the neighbor list is permuted; default $= 5$.
- **Top-k Parameter** k: Number of co-activated nodes to be selected; default $=$ average degree.
- **Hub Threshold** D: A fixed degree cutoff that determines whether an activated node is allowed to propagate the contagion based on its degree; default $=$ median degree.

TAS activates nodes only when they receive sufficient multi-neighbor reinforcement, governed by a global structural threshold τ. TAS uses the same parameters as MAS, with the addition of a threshold set $\mathcal{T}$ that determines the values τ iterates over. By default, $\mathcal{T} = \{1, 2, 3, 4, 5\}$.

Seed-Aware Hub-Restricted Propagation. We distinguish between using a hub as a seed and using a hub as a propagation source. For a seed node v, the algorithm may access $N(v)$ to form the initial subset $R \subseteq N(v)$; this cost is charged once per permutation of the seed neighborhood. After initialization, however, an activated node u is expanded only if $d(u) \leq D$, where D is a fixed degree threshold. If $d(u) > D$, then u remains active and contributes to the co-activation counts, but its adjacency list is not scanned and the contagion does not propagate through it. Thus, when a hub is the seed, repeated chunks of $N(v)$ still probe the different regions incident to the hub, while cascades seeded elsewhere do not expand explosively by propagating through a hub reached later in the process.

Sparse Counting/Top-k Selection. Although the pseudocode may suggest a counter over all nodes, the implementation maintains a *sparse* associative map (hash/dictionary) that stores counts only for nodes that appear in cascades. The top-k set is computed over these nonzero entries via a heap, rather than by sorting all n nodes.

Constant-Parameter Regime. In our experiments, $P, \ell, r, k,$ and $|\mathcal{T}|$ are treated as constants independent of n and m (i.e., small fixed hyperparameters). We use adjacency lists to iterate over the neighbors of a node u, and the associated complexity is $O(\deg(u))$.

C.2 Time Complexity

Seed-Neighborhood Initialization Cost. For a source node v, the algorithm accesses $N(v)$ only to form the initial subsets $R \subseteq N(v)$. Let

$$q_v := \max\left\{1, \left\lceil \frac{d(v)}{r} \right\rceil \right\}$$

be the number of neighbor chunks used for v. Across one permutation of $N(v)$, forming all chunks costs $O(d(v))$. Hence, across P permutations and $|\mathcal{T}|$ thresholds, the total seed-neighborhood initialization cost is

$$O\left(P|\mathcal{T}| \sum_{v \in V} d(v)\right) = O(P|\mathcal{T}|m).$$

For fixed P and $|\mathcal{T}|$, this is $O(m)$.

Algorithm 1 Maximum Adjacency Search (MAS) with Seed-Aware Hub-Restricted Propagation

Require: Graph $G = (V, E)$, walk length ℓ, number of starting neighbors r, number of permutations P, top-k parameter k, hub threshold D
Ensure: Dictionary $\mathcal{F}$ storing the top-k co-activation counts for each seed node

$\mathcal{F} \leftarrow$ empty dictionary
for $v \in V$ **do**
  $N_v \leftarrow N(v)$ ▷ seed-neighborhood access
  counter $\leftarrow$ empty sparse map
  if $|N_v| > 0$ **then**
    for $p = 1, \ldots, P$ **do**
      $\pi_v \leftarrow \mathrm{permute}(N_v)$
      for $i \in \{0, r, 2r, \ldots, |N_v| - 1\}$ **do**
        $s \leftarrow \min\{i, \max(0, |N_v| - r)\}$
        $R \leftarrow \pi_v[s : s + r]$
        $S \leftarrow \{v\} \cup R$ ▷ initial active set
        support $\leftarrow$ empty sparse map
        processed $\leftarrow \emptyset$
        $Q \leftarrow \{x \in S : d(x) \le D\}$
        while $Q \neq \emptyset$ **do**
          $x \leftarrow \mathrm{pop}(Q)$
          if $x \in$ processed **then**
            continue
          end if
          processed $\leftarrow$ processed $\cup \{x\}$
          for $y \in N(x)$ **do**
            if $y \notin S$ **then**
              $\mathrm{support}[y] \leftarrow \mathrm{support}[y] + 1$
            end if
          end for
        end while
        for $t = 1, \ldots, \ell$ **do**
          if there is no $y \notin S$ with $\mathrm{support}[y] > 0$ **then**
            break
          end if
          $u \leftarrow \arg\max_{y \notin S} \mathrm{support}[y]$
          $S \leftarrow S \cup \{u\}$
          delete $\mathrm{support}[u]$
          if $d(u) \le D$ **then**
            for $y \in N(u)$ **do**
              if $y \notin S$ **then**
                $\mathrm{support}[y] \leftarrow \mathrm{support}[y] + 1$
              end if
            end for
          end if ▷ if $d(u) > D$, then u is active but not expanded
        end for
        for $u \in S \setminus \{v\}$ **do**
          $\mathrm{counter}[u] \leftarrow \mathrm{counter}[u] + 1$
        end for
      end for
    end for
  end if
  $\mathcal{F}[v] \leftarrow \mathrm{TopK}(\mathrm{counter}, k)$ ▷ select top-k
end for
return $\mathcal{F}$

Number of Cascade Initializations. The number of chunks for source v is q_v, so

$$\sum_{v \in V} q_v \le \sum_{v \in V} \left(1 + \frac{d(v)}{r}\right) = n + \frac{2m}{r}.$$

Algorithm 2 Threshold Adjacency Search (TAS) with Seed-Aware Hub-Restricted Propagation

Require: Graph $G = (V, E)$, walk length ℓ, number of starting neighbors r, number of permutations P, threshold set $\mathcal{T}$, top-k parameter k, hub threshold D
Ensure: Dictionary $\mathcal{F}$ storing the top-k co-activation counts for each source node

$\mathcal{F} \leftarrow$ empty dictionary
for $v \in V$ **do**
  $N_v \leftarrow N(v)$ ▷ seed-neighborhood access
  counter $\leftarrow$ empty sparse map
  if $|N_v| > 0$ **then**
    for $\tau \in \mathcal{T}$ **do**
      for $p = 1, \ldots, P$ **do**
        $\pi_v \leftarrow \mathrm{permute}(N_v)$
        for $i \in \{0, r, 2r, \ldots, |N_v| - 1\}$ **do**
          $s \leftarrow \min\{i, \max(0, |N_v| - r)\}$
          $R \leftarrow \pi_v[s : s + r]$
          $S_0 \leftarrow \{v\} \cup R$
          $S \leftarrow$ HUBRESTRICTED-DELAYED-BFS(G, S_0, ℓ, τ, D)
          for $u \in S \setminus \{v\}$ **do**
            $\mathrm{counter}[u] \leftarrow \mathrm{counter}[u] + 1$
          end for
        end for
      end for
    end for
  end if
  $\mathcal{F}[v] \leftarrow \mathrm{TopK}(\mathrm{counter}, k)$ ▷ select top-k
end for
return $\mathcal{F}$

Therefore, MAS runs at most $P\left(n + \frac{2m}{r}\right)$ cascades and TAS runs at most $P|\mathcal{T}|\left(n + \frac{2m}{r}\right)$ cascades.

Hub-Restricted Propagation Cost. Each cascade starts from at most $r + 1$ active nodes and activates at most ℓ additional nodes. Thus, each cascade processes at most $r + 1 + \ell$ active nodes. After initialization, an active node u is expanded only if $d(u) \le D$. If $d(u) > D$, then u is counted but $N(u)$ is not scanned. Hence, every expanded node contributes at most D adjacency inspections, and every cascade has propagation cost

$$O(D(r + \ell)).$$

For fixed D, r, ℓ, this is constant per cascade. Combining with the cascade count above gives total propagation cost

$$O\left(P|\mathcal{T}|D(r + \ell)\left(n + \frac{2m}{r}\right)\right) = O(n + m)$$

when $P, |\mathcal{T}|, D, r, \ell$ are fixed constants. In the experiments D is set to the median degree.

Sparse Counting and Top-k Extraction. For each seed v, each cascade contributes at most $r+1+\ell$ activated nodes to the sparse co-activation counter. Therefore, the total number of nonzero counter updates over all seeds is

$$O\left(P|\mathcal{T}|(r + \ell)\sum_{v \in V} q_v\right) = O(n + m)$$

under fixed cascade parameters.

Let C_v be the set of nodes with a nonzero co-activation count for seed v. Using linear-time selection, the top-k nodes can be extracted in $O(|C_v| + k)$ time. Thus, top-k extraction over all seeds costs

$$O\left(\sum_v |C_v| + nk\right) = O(n + m + nk).$$

If k is fixed, or if k is set to the rounded average degree $k = O(1 + m/n)$ as in our experiments, then $nk = O(n + m)$.

Algorithm 3 HUBRESTRICTED-DELAYED-BFS

Require: Graph $G = (V, E)$, initial active set S_0, walk length ℓ, threshold τ, hub threshold D
Ensure: Activated set S

$S \leftarrow S_0$
support $\leftarrow$ empty sparse map
processed $\leftarrow \emptyset$
$I \leftarrow$ queue initialized with all $x \in S_0$ such that $d(x) \le D$
new $\leftarrow 0$
while $I \neq \emptyset$ and new $< \ell$ **do**
  $x \leftarrow \text{dequeue}(I)$
  if $x \in$ processed **then**
    continue
  end if
  processed $\leftarrow$ processed $\cup \{x\}$
  for $y \in N(x)$ **do**
    if $y \notin S$ **then**
      support$[y] \leftarrow$ support$[y] + 1$
      if support$[y] \ge \tau$ **then**
        $S \leftarrow S \cup \{y\}$
        new $\leftarrow$ new $+ 1$
        delete support$[y]$
        if $d(y) \le D$ **then**
          enqueue(I, y)
        end if ▷ if $d(y) > D$, then y is active but not expanded
        if new $= \ell$ **then**
          break
        end if
      end if
    end if
  end for
end while
return S

Overall Complexity. Under seed-aware hub-restricted propagation, fixed $P, |\mathcal{T}|, r, \ell, D$, sparse counters, and $k = O(1 + m/n)$, the total construction time of $G^* = (V, E^*, W^*)$ is

$$O(n + m).$$

The output size is also bounded by

$$|E^*| \le nk = O(n + m),$$

and when k is set to the average degree, $|E^*| = O(m)$.

Without the hub-restricted propagation rule, the full-exposure implementation has output-sensitive runtime

$$O\left(\sum_{v \in V} \sum_{c \in \mathcal{C}(v)} \sum_{u \in A_{v,c}} d(u)\right),$$

which can be superlinear on graphs where high-degree hubs are repeatedly reached from many seeds. Hub-restricted propagation replaces these repeated hub scans by $O(1)$ counting operations, yielding the worst-case linear bound above.

D Further Experiments and Implementation Details

This section provides implementation details and additional experimental results that complement the main results of the paper. We focus on robustness analysis, parameter sensitivity, and extended

evaluations across graph structures to better understand the behavior of the proposed methods. The main results are presented in Section 5, with standard deviations provided in Appendix D.3.

D.1 Implementation Details

We evaluate the GNN and GT contagion-based models using two layers with a hidden dimension of 512, a dropout rate of 10%, and all other model parameters set to their default. Attention-based architectures (GAT and all GTs) additionally employ 8 attention heads. In GNN architectures, ReLU activation is applied between layers before dropout, and log-softmax activation is applied at the output layer.

All experiments are implemented in PyTorch and PyTorch Geometric. All training is performed using the Adam optimizer with a peak learning rate of 0.01 and weight decay of 1×10^{-5}, minimizing the negative log-likelihood loss. We employ a linear learning rate scheduler with 500 warm-up steps, followed by linear decay from 0.01 to 0.0001 over 1000 total training steps. Each model is trained for up to 2000 epochs using a batch size of 2000. Early stopping is applied with a patience of 50 epochs.

Hardware. All experiments were conducted on a workstation equipped with a single AMD Ryzen Threadripper PRO 5955WX processor (16 cores, 4.00–4.50GHz, 64MB cache, PCIe 4.0), one NVIDIA GeForce RTX 4090 GPU, and 128GB of DDR4-3200.

Table 7: DGL benchmark dataset statistics.

Dataset	# Nodes	# Edges	# Labels	# Features	Average Degree	Label Homophily	Connected
actor	7600	29707	5	932	7.82	**0.22**	Yes
chameleon	2277	31421	5	2325	27.60	**0.23**	Yes
citeseer	3327	4676	6	3703	2.81	0.74	No
community	1400	3871	8	10	5.53	0.70	Yes
computer	13752	245861	10	767	35.76	0.78	No
cora	2708	5278	7	1433	3.90	0.81	No
cornell	183	280	5	1703	3.06	**0.13**	Yes
cycle	871	970	2	1	2.23	0.90	Yes
grid	1231	1705	2	1	2.77	0.91	Yes
photo	7650	119082	8	745	31.13	0.83	No
pubmed	19717	44327	3	500	4.50	0.80	Yes
shape	700	1760	4	1	5.03	0.77	Yes
squirrel	5201	198493	5	2089	76.33	**0.22**	Yes
texas	183	295	5	1703	3.22	**0.11**	Yes
wiki	11701	216123	10	300	36.94	0.66	No
wisconsin	251	466	5	1703	3.71	**0.21**	Yes

D.2 Baseline Methods and Summary of Findings

GNN and GT baselines. We compare with 16 representative approaches: GCN [Kipf and Welling, 2017], GraphSAGE [Hamilton et al., 2017], GAT [Veličković et al., 2018], PPRGo [Bojchevski et al., 2020], GRAND+ [Feng et al., 2022], GT [Dwivedi and Bresson, 2020], Gophormer [Zhao et al., 2021], Graphormer [Ying et al., 2021], SAN [Kreuzer et al., 2021], GraphGPS [Rampášek et al., 2022], NAGphormer [Chen et al., 2023]NAG, Specformer [Bo et al., 2023], Exphormer [Shirzad et al., 2023], SGFormer [Wu et al., 2023], VCR-Graphormer [Fu et al., 2024]VCR, and PolyFormer [Ma et al., 2024]. Across datasets, GRAPH CASCADES consistently enhances node classification across various GNN and GT backbones, achieving competitive performance and largest gains on heterophilic graphs as well as homophilic graphs with moderate-to-high degrees (see Section 5, Appendices D.3-D.5, and Appendices D.9-D.10).

Rewiring Baselines. To position GRAPH CASCADES against existing graph rewiring methods, we compare cascade operators (MAS and TAS) against structural baselines targeting oversquashing (FOSR [Karhadkar et al., 2023] and SDRF [Topping et al., 2022]), a diffusion-based sparsification method (GDC [Gasteiger et al., 2019]), and a direct two-hop common-neighbor proxy (A^2 rewiring). By adding edges that increase algebraic connectivity (FOSR) or break high-curvature bottlenecks

(SDRF), these structural methods *create* shortcuts across narrow cuts to alleviate bottleneck-induced oversquashing. Meanwhile, GDC sparsifies global diffusion matrices to smooth feature propagation, and A^2 rewiring relies on static two-hop common-neighbor counts to reinforce local node overlap. In contrast, GRAPH CASCADES is complementary rather than competitive: it *reinforces* mesoscopic regions using dynamic, weighted multi-hop flux that already contains redundant short-path support, yet provably cannot bridge cuts of size smaller than κ (Appendix B.10). These approaches therefore span complementary structural philosophies: bottleneck-targeted methods (FOSR, SDRF) excel on graphs dominated by narrow cuts, while diffusion and local overlap baselines (GDC, A^2)—alongside GRAPH CASCADES—operate primarily on graphs whose label-relevant structure benefits from multi-hop neighborhood reinforcement. The empirical comparison in Appendix D.7 confirms this division.

Heterophily Baselines. To assess how GRAPH CASCADES compares against architectures explicitly designed for heterophilic graphs, we run additional experiments against seven state-of-the-art baselines: ACMGCN [Luan et al., 2022], CMGNN [Zheng et al., 2025], GGCN [Yan et al., 2022], GloGNN [Li et al., 2022], H2GCN [Zhu et al., 2020], M2MGNN [Liang et al., 2024], and OrderedGNN [Song et al., 2023]. These methods incorporate heterophily-specific mechanisms such as adaptive channel mixing, compatibility-matrix-aware message passing, multiset-to-multiset aggregation, and ordered hop-wise neuron blocks, and have been shown to outperform standard GNNs on heterophilic benchmarks. Overall, GRAPH CASCADES remains competitive with specialized heterophily-aware architectures, achieving top-tier performance across diverse graph regimes without requiring architecture-specific modifications. Further details are provided in Section D.8.

D.3 Main Results with Standard Deviations

Table 8: Test accuracy (mean ± std, in %) across DGL datasets. "Our Best Validation selected Method" indicates the best validation-selected cascade models from Table 9 per dataset. OOM indicates out-of-memory errors. Accuracies for the top-5 best-performing baselines are encoded by shading, with darker cells indicating better performance.

		actor	***chameleon***	***citeseer***	***community***	***computer***	***cora***	***cornell***	***cycle***
GNNs	GCN	29.59±0.91	57.30±2.67	75.70±1.41	55.63±1.92	90.30±0.66	87.96±0.83	45.11±6.20	60.73±6.59
	GraphSAGE	34.57±1.03	64.36±1.60	75.88±1.44	52.09±2.06	90.39±0.50	88.09±0.89	64.47±4.69	59.27±2.73
	GAT	28.85±0.95	58.63±2.57	75.77±1.37	58.99±2.57	OOM	87.38±1.02	44.15±6.40	59.27±2.73
	PPRGo	31.36±1.14	45.83±2.62	74.33±0.91	39.43±2.19	91.13±0.58	84.30±0.73	46.91±7.09	59.27±2.73
	GRAND+	30.20±1.08	45.54±2.73	73.80±0.60	39.37±2.01	90.85±0.69	83.70±0.46	45.11±6.46	59.27±2.73
GTs	GT	34.19±1.16	64.83±1.56	68.60±5.85	54.00±3.45	OOM	85.41±0.81	61.91±8.02	56.32±7.47
	Gophormer	34.90±1.40	53.16±3.32	31.48±9.62	71.00±4.17	90.60±0.68	49.18±12.26	67.13±6.92	54.98±9.41
	Graphormer	35.01±1.09	50.70±4.40	33.18±2.97	73.20±2.30	OOM	35.89±5.59	51.49±9.60	81.48±6.13
	SAN	36.04±0.91	59.22±2.00	73.37±2.73	46.00±2.39	OOM	85.07±1.47	67.55±5.69	59.27±2.73
	GraphGPS	35.19±1.31	60.78±1.35	75.23±1.63	44.00±2.30	OOM	86.65±1.04	61.38±5.49	55.11±10.10
	NAG	31.68±1.26	40.38±1.81	74.72±1.67	46.17±2.49	88.66±0.73	86.75±0.82	58.94±7.11	77.24±9.34
	Specformer	30.74±1.10	42.29±2.45	70.33±1.78	47.69±3.84	88.05±0.39	82.49±1.08	54.00±3.90	54.32±1.30
	Exphormer	36.62±0.89	59.36±3.10	68.34±2.11	65.63±2.75	89.22±0.80	78.17±2.40	68.19±6.75	61.80±6.57
	SGFormer	31.06±1.33	42.37±3.90	69.91±1.11	44.61±2.76	85.40±0.23	81.41±2.80	55.19±2.43	55.19±1.44
	VCR	26.41±1.20	27.28±2.51	71.15±1.36	28.13±4.60	82.45±5.30	85.41±1.33	44.89±6.09	61.64±4.12
	PolyFormer	32.12±1.12	46.83±2.61	72.13±1.22	48.91±1.77	89.60±0.31	83.51±2.17	59.02±3.67	58.11±2.91
Our Validated	**Best Method**	GPS-TAS 36.46±1.31	CR-TAS 63.65±2.82	GPS-TAS 75.28±1.31	CR-TAS 78.60±3.50	GCN-TAS 90.87±0.79	GCN-TAS 87.29±0.77	NAG-MAS/TAS 74.36±5.96	CR-TAS 61.32±3.67

		grid	***photo***	***pubmed***	***shape***	***squirrel***	***texas***	***wiki***	***wisconsin***
GNNs	GCN	59.34±5.45	93.90±0.74	87.16±0.35	89.46±11.87	39.10±1.43	54.57±7.12	83.89±0.73	51.41±4.81
	GraphSAGE	58.19±1.76	95.31±0.60	87.33±0.34	43.11±2.97	45.41±1.19	79.79±6.19	84.57±0.78	72.89±4.63
	GAT	58.19±1.76	94.49±0.52	86.88±0.45	43.11±2.97	OOM	56.70±6.45	OOM	49.45±5.54
	PPRGo	58.19±1.76	95.09±0.63	86.72±0.39	43.11±2.97	30.22± 1.56	58.51±5.46	84.16±0.98	55.63±4.90
	GRAND+	58.19±1.76	94.82±0.59	85.95±0.43	43.11±2.97	30.23±1.54	56.49±6.34	83.38±0.89	53.20±5.33
GTs	GT	54.64±7.14	OOM	86.59±0.79	23.23±2.73	OOM	77.77±5.05	OOM	73.52±4.49
	Gophormer	62.62±8.81	94.98±0.64	40.40±13.63	79.83±8.97	37.44±2.80	81.49±5.82	83.34±0.91	81.95±5.08
	Graphormer	72.14±2.65	89.02±0.75	OOM	85.57±2.15	40.55±2.05	68.62±10.28	OOM	71.33±8.80
	SAN	57.27±4.26	94.85±1.24	OOM	29.40±9.52	39.00±1.38	80.00±7.48	OOM	80.63±3.91
	GraphGPS	57.41±10.00	94.64±0.63	OOM	46.74±14.93	40.04±1.27	69.68±6.58	OOM	72.66±4.95
	NAG	70.99±1.99	93.94±0.67	87.26±0.52	49.89±6.51	25.80±1.26	63.30±6.43	82.40±0.98	61.25±4.32
	Specformer	53.77±1.19	89.22±0.76	85.58±0.48	45.77±6.69	31.44±2.30	63.26±5.32	71.82±0.31	68.90±5.01
	Exphormer	57.31±4.95	94.25±0.65	86.97±0.61	76.31±3.44	42.31±3.37	79.47±6.95	83.84±0.86	80.39±5.56
	SGFormer	52.53±3.77	87.93±1.14	85.07±0.81	42.80±2.10	32.86±3.29	61.61±5.01	67.03±0.39	63.86±4.38
	VCR	58.32±1.96	88.45±7.85	83.25±0.44	44.66±6.15	22.48±2.00	54.68±7.28	78.46±2.69	51.33±7.46
	PolyFormer	56.15±4.23	91.93±0.21	86.82±0.17	47.29±8.02	34.42±4.45	64.38±4.40	73.18±0.99	69.77±3.73
Our Validated	**Best Method**	NAG-MAS 62.25±2.21	GPS-TAS 94.84±0.82	VCR-TAS 88.70±0.91	GCN-TAS 89.57±11.79	CR-TAS 47.31±2.19	VCR-TAS 82.31±6.59	GCN-TAS 83.54±0.76	NAG-MAS/TAS 81.02±4.13

Table 9: Test accuracy (mean ± std, in %) across DGL datasets. OOM indicates out-of-memory errors. Darker cell colors indicate better performance.

Method	*actor*	*chameleon*	*citeseer*	*community*	*computer*	*cora*	*cornell*	*cycle*
GCN	29.59±0.91	57.30±2.67	75.70±1.41	55.63±1.92	90.30±0.66	87.96±0.83	45.11±6.20	60.73±6.59
GCN-MAS	34.28±0.96	57.67±2.07	74.71±1.61	51.76±2.06	89.34±0.84	84.93±1.12	60.85±5.00	59.27±2.73
GCN-TAS	34.62±0.92	58.75±2.54	75.46±1.46	50.11±2.36	90.87±0.79	87.29±0.77	57.02±5.37	60.39±3.76
GraphGPS	35.19±1.31	60.78±1.35	75.23±1.63	44.00±2.30	OOM	86.65±1.04	61.38±5.49	55.11±10.10
GPS-MAS	36.39±1.17	59.90±1.75	74.09±1.38	47.43±3.04	OOM	85.49±1.46	68.72±6.02	55.78±2.70
GPS-TAS	36.46±1.31	59.66±1.66	75.28±1.31	45.26±2.95	OOM	86.53±1.08	67.23±6.49	57.51±5.41
NAG	31.68±1.26	40.38±1.81	74.72±1.67	46.17±2.49	88.66±0.73	86.75±0.82	58.94±7.11	77.24±9.34
NAG-MAS	35.10±1.03	59.15±1.79	72.17±1.61	52.59±2.21	89.10±0.59	84.84±0.74	74.36±5.96	59.00±2.86
NAG-TAS	35.10±1.03	58.41±2.26	74.15±1.63	58.21±2.86	89.19±0.69	86.82±1.00	74.36±5.96	59.20±2.19
VCR	26.41±1.20	27.28±2.51	71.15±1.36	28.13±4.60	82.45±5.30	85.41±1.33	44.89±6.09	61.64±4.12
VCR-MAS	36.18±1.23	60.02±2.87	71.47±1.64	47.22±2.98	89.43±2.43	85.08±0.57	73.85±5.72	51.70±3.31
VCR-TAS	36.28±1.42	59.61±1.95	74.14±1.53	57.97±1.81	88.95±2.39	85.11±0.62	75.02±4.68	50.09±4.37
CR-MAS	31.52±1.47	62.82±2.06	50.63±2.29	54.04±2.17	89.44±0.69	79.81±2.01	54.04±7.38	59.18±2.55
CR-TAS	34.64±1.21	63.65±2.82	51.84±4.17	78.60±3.50	89.33±0.82	81.02±2.05	55.74±6.85	61.32±3.67

Method	*grid*	*pubmed*	*photo*	*shape*	*squirrel*	*texas*	*wiki*	*wisconsin*
GCN	59.34±5.45	93.90±0.74	87.16±0.35	89.46±11.87	39.10±1.43	54.57±7.12	83.89±0.73	51.41±4.81
GCN-MAS	58.30±1.84	93.91±0.63	87.87±0.54	57.69±5.49	40.01±1.96	66.49±7.71	81.79±0.66	68.52±4.82
GCN-TAS	58.19±1.76	94.17±0.60	88.15±0.45	89.57±11.79	42.38±1.68	65.00±6.42	83.54±0.76	71.56±4.41
GraphGPS	57.41±10.00	94.64±0.63	OOM	46.74±14.93	40.04±1.27	69.68±6.58	OOM	72.66±4.95
GPS-MAS	55.44±6.50	94.71±0.78	OOM	54.89±6.41	39.20±1.17	75.00±5.56	OOM	78.44±3.68
GPS-TAS	56.33±5.59	94.84±0.82	OOM	56.26±13.41	39.18±1.34	75.64±4.71	OOM	78.28±3.32
NAG	70.99±1.99	93.94±0.67	87.26±0.52	49.89±6.51	25.80±1.26	63.30±6.43	82.40±0.98	61.25±4.32
NAG-MAS	62.25±2.21	94.27±0.69	88.19±0.45	57.54±4.31	40.89±1.60	81.28±6.16	82.38±1.19	81.02±4.13
NAG-TAS	60.60±2.44	94.61±0.51	88.26±0.37	84.43±4.05	42.13±1.41	81.28±6.16	82.59±1.20	81.02±4.13
VCR	58.32±1.96	88.45±7.85	83.25±0.44	44.66±6.15	22.48±2.00	54.68±7.28	78.46±2.69	51.33±7.46
VCR-MAS	44.05±2.58	93.88±0.29	87.59±0.46	52.12±5.17	39.37±1.49	80.99±5.63	82.68±0.54	79.93±3.76
VCR-TAS	47.84±1.28	94.22±1.27	88.70±0.91	76.94±3.73	46.06±1.04	82.31±6.59	82.84±0.52	81.00±2.66
CR-MAS	61.80±2.51	93.73±0.89	87.53±0.63	54.37±5.11	45.72±2.69	65.43±9.02	82.65±0.90	66.56±5.20
CR-TAS	60.74±3.46	93.95±0.52	87.75±0.56	85.94±7.79	47.31±2.19	65.43±6.79	83.04±0.83	70.39±5.33

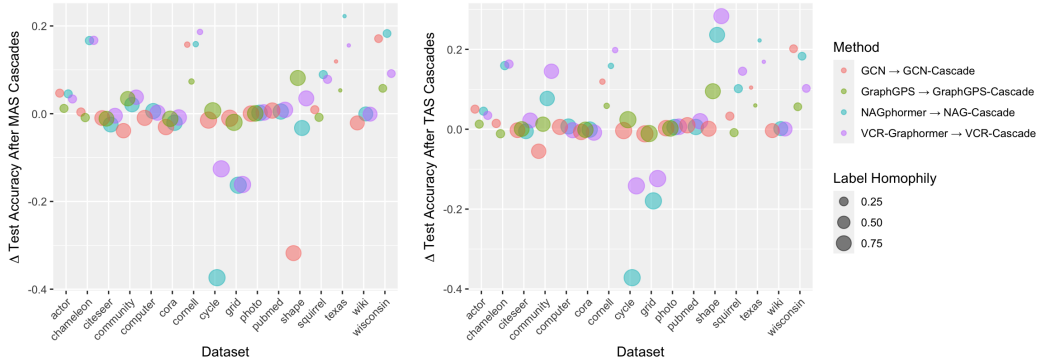

Figure 1: Changes in mean test accuracy on DGL graphs as a result of MAS and TAS cascades.

D.4 Effect of Cascades on Homophily

Main Observations. Figure 2 shows that, after rewiring, low-degree homophilic graphs become markedly less homophilic, whereas moderate- to high-degree homophilic and heterophilic graphs become mildly more homophilic. Benchmark graphs exhibit the same trend as the synthesized ones, consistent with Proposition 4.5.

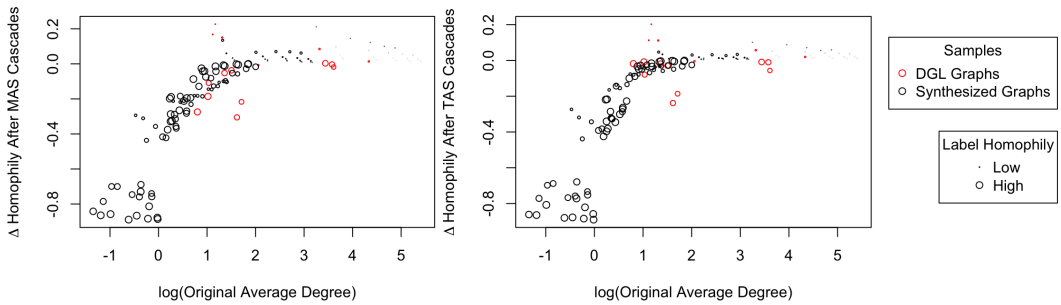

Figure 2: Effects of mesoscopic cascades on label homophily.

Empirical Verification of Proposition 4.5 Table 10 reports the resulting homophily quantities h_G and $h_{\mathcal{F}}$, which are the conditional probabilities most directly relevant to the classification effect, computed using the population reinforcement event $\mathcal{F}$ from Proposition 4.5 with $\ell \in \{2, 4, 6\}$ and $\kappa = 2$. The table organizes datasets by which assumptions of Proposition 4.5 are expected to hold, and pairs the homophily diagnostic with the maximum within-architecture accuracy gain from cascade rewiring (Δ_{acc}). Three regimes emerge.

On heterophilic graphs (*texas*, *cornell*, *wisconsin*, *chameleon*, *squirrel*, *actor*), both Assumptions 4.3 and 4.4 hold, $h_{\mathcal{F}}$ exceeds h_G by a factor of 1.5–4.2, and rewiring yields large accuracy gains ($+9.87$ to $+32.74$). On moderate- to high-degree homophilic graphs (*community*, *shape*, *computer*, *photo*, *wiki*, *pubmed*), Assumption 4.4 fails because reinforcement is no longer selective relative to direct adjacency; consequently, $h_{\mathcal{F}}$ drops slightly below h_G, and the mechanism predicts no homophily-driven gain. We still observe positive Δ_{acc} here, indicating that on these graphs the rewiring contributes through a complementary mechanism (mesoscopic structural reorganization, c.f. Theorem 4.8) rather than through homophily amplification. On low-degree highly homophilic graphs (*citeseer*, *cora*, *cycle*, *grid*), Assumption 4.3 fails: same-label pairs are not connected by enough short walks to be reinforced, so $h_{\mathcal{F}}$ falls substantially below h_G and rewiring yields no benefit. The failure is most pronounced on the highly regular cycle and grid graphs, where the sparse and uniform structure prevents reinforcement from concentrating on informative neighbors. On *grid*, all the baselines (see Table 8) degrade under cascade rewiring, with within-architecture losses ranging from -1.04 to -10.48, so $\Delta_{\text{acc}} \leq 0$ across the board. On *cycle*, three of the four backbones lose substantially (-0.34, -18.04, -11.55); the single positive value ($\Delta_{\text{acc}} = +2.40$, GraphGPS) is therefore an outlier rather than a representative gain, so we mention $\Delta_{\text{acc}} \leq 0$. The diagnostic therefore calibrates the *direction* of the rewiring effect across all sixteen benchmarks consistently with Proposition 4.5: gains where both assumptions hold, neutrality where Assumption 4.4 fails, and degradation where Assumption 4.3 fails.

Table 10: Empirical verification of the reinforcement homophily mechanism (Proposition 4.5) across DGL benchmarks. Rows are grouped by direct evaluation of the alignment and selectivity inequalities $\Pr[\mathcal{F} \mid S] \geq \Pr[\mathcal{E} \mid S]$ and $\Pr[\mathcal{F}] \leq \Pr[\mathcal{E}]$; the raw probabilities are omitted for readability. $h_G = \Pr[y_u = y_v \mid (u, v) \in E]$ is the original-graph edge homophily; $h_{\mathcal{F}} = \Pr\left[y_u = y_v \mid (A^2 + \cdots A^\ell)_{uv} \geq \kappa\right]$ is the population reinforcement homophily, computed by direct enumeration with $\ell \in \{2, 4, 6\}$, $\kappa = 2$. Δ_{acc} reports the maximum within-architecture gain from cascade rewiring across baselines, computed from Table 8.

Dataset	h_G	$h_{\mathcal{F}}$	$h_{\mathcal{F}} - h_G$	$h_{\mathcal{F}}/h_G$	Δ_{acc}
Heterophilic — both Assumptions 4.3 and 4.4 hold ⇒ Proposition 4.5 applies					
cornell	0.13	0.55	$+0.42$	$4.2\times$	$+30.13$
texas	0.11	0.42	$+0.31$	$3.8\times$	$+27.63$
wisconsin	0.21	0.48	$+0.27$	$2.3\times$	$+29.67$
chameleon	0.23	0.46	$+0.23$	$2.0\times$	$+32.74$
squirrel	0.22	0.33	$+0.12$	$1.6\times$	$+23.58$
actor	0.22	0.32	$+0.10$	$1.5\times$	$+9.87$
Mid-Degree Homophilic — Assumption 4.4 fails mildly; $h_F \lesssim h_G$					
community	0.70	0.65	-0.05	$0.93\times$	$+29.84$
shape	0.77	0.70	-0.07	$0.91\times$	$+34.54$
pubmed	0.80	0.70	-0.10	$0.88\times$	$+5.45$
High-Degree Homophilic — Assumption 4.4 fails moderately due to noise dilution					
photo	0.83	0.70	-0.13	$0.84\times$	$+5.77$
computer	0.78	0.65	-0.13	$0.83\times$	$+6.98$
wiki	0.66	0.55	-0.11	$0.83\times$	$+4.38$
Low-Degree Homophilic — Assumption 4.3 fails as same-label pairs lack walk support					
citeseer	0.74	0.55	-0.19	$0.74\times$	$+2.99$
cora	0.81	0.60	-0.21	$0.74\times$	$+0.07$
cycle	0.90	0.55	-0.35	$0.61\times$	≤ 0
grid	0.91	0.60	-0.31	$0.66\times$	≤ 0

Shortcut Quality and Node-Level Prediction Changes. Here we analyze the *actual nonlocal shortcut neighborhoods* added by GRAPH CASCADES and relate them directly to node-level shifts in

the true-class log-probability margin,

$$\Delta m_v = m_v^{\text{cascade}} - m_v^{\text{baseline}},$$

and separate prediction changes into:

- **wrong-to-right (WR)**: errors corrected by rewiring.
- **right-to-wrong (RW)**: new errors introduced by rewiring.

Thus, WR and RW measure beneficial changes and injected prediction noise separately rather than combining them into a single net-accuracy value. Labels are used only for this post-hoc analysis; neither the rewiring procedure nor the model uses them.

Table 11: Node-level prediction change analysis (wrong-to-right vs. right-to-wrong) and mean log-probability margin changes (Δm_v) using NAGphormer as the backbone across graph regimes.

Regime	**Dataset**	n_{test}	**WR : RW (MAS)**	**WR : RW (TAS)**	**Mean Δm_v (MAS / TAS)**
Heterophilic	chameleon	570	$195 : 78 = \mathbf{2.50}\times$	$203 : 95 = \mathbf{2.14}\times$	$+0.60/+0.57$
	texas	47	$14 : 12 = 1.17\times$	$13 : 8 = 1.62\times$	$+0.12/+0.46$
	wisconsin	64	$15 : 11 = 1.36\times$	$13 : 12 = 1.08\times$	$+0.51/+0.05$
	actor	1900	$425 : 402 = 1.06\times$	$417 : 388 = 1.07\times$	$+0.02/+0.06$
	cornell	47	$13 : 10 = 1.30\times$	$10 : 14 = 0.71\times$	$-0.12/-0.49$
Mid to high-degree	community	350	$102 : 84 = 1.21\times$	$111 : 74 = 1.50\times$	$+0.48/+0.60$
homophilic	shape	175	$55 : 37 = 1.49\times$	$83 : 15 = \mathbf{5.53}\times$	$-0.23/+0.82$
Low-degree	cora	677	$24 : 42 = 0.57\times$	$22 : 14 = 1.57\times$	$-0.14/-0.07$
homophilic	citeseer	833	$154 : 183 = 0.84\times$	$165 : 169 = 0.98\times$	$-0.28/-0.15$
Bottlenecked	grid	309	$55 : 70 = 0.79\times$	$56 : 76 = 0.74\times$	$-0.27/-0.23$
or regular	cycle	219	$17 : 77 = \mathbf{0.22}\times$	$21 : 79 = \mathbf{0.27}\times$	$-2.65/-2.66$

Table 11 reports the analysis with NAGphormer as the backbone. The node-level pattern closely follows the regimes identified in Table 5. Among the 14 dataset–operator pairs in the heterophilic and moderate to high-degree homophilic regimes, *13 produce more corrected predictions than newly introduced errors*, and *11 also increase the mean true-class margin*. The strongest examples are *chameleon*, where MAS and TAS correct $2.50\times$ and $2.14\times$ as many predictions as they corrupt, and *shape*, where TAS reaches a $5.53\times$ ratio.

The predicted failure regimes provide a useful negative control. In seven of the eight low-degree or bottlenecked pairs, WR $\leq$ RW, and all eight have negative mean Δm_v. The clearest example is *cycle*: MAS and TAS introduce approximately four times as many errors as they correct, consistent with the lack of redundant short-path support identified by Proposition B.10 and Corollary B.11.

Overall, the analysis shows that cascade rewiring does not merely raise aggregate graph homophily: in the regimes where it is designed to operate, the added shortcut neighborhoods predominantly *correct existing errors, increase prediction margins, and introduce substantially fewer new errors*.

D.5 Correlation Between Classification Accuracy and Structural Properties

Table 2 suggests a correlation between the classification accuracy of tokenized sparse-attention GTs and graph structural properties, including label homophily, the average degree of the auxiliary graph, and the connectivity of the original graph. Notably, the correlations obtained from cascade models are clearer than those from the baseline GTs. Since these patterns are observed on the combined set of synthesized and benchmark graphs—and the synthesized graphs dominate in number—an important question is whether the same correlations persist independently within benchmark graphs.

Due to the imbalance in the homophily distribution of the benchmark graphs, performing separate regressions on the synthesized and benchmark graph sets is not feasible. Instead, we plot test accuracy against the input graph's label homophily and encode additional structural properties—the average degree of the input graph (original graph for NAG and VCR, and auxiliary graph for the cascade models), the connectivity of the original graph, and whether the graph is a DGL benchmark—using marker size, shape, and color, respectively. Figure 3 illustrates the interaction among these structural factors, consistent with the regression results in Table 2; the patterns observed in the DGL benchmark graphs closely align with those of the synthesized graphs.

D.6 Ablation Study

In this section, we present a parameter analysis of CR-GRAPHORMER and justify the use of cascade frequency-based edge weights W^* through additional classification results.

D.6.1 Parameter Analysis for CR-GRAPHORMER

We perform a sensitivity analysis with respect to the main hyperparameters controlling the MAS and TAS cascades, using CR-GRAPHORMER as the cascade model. In this experiment, we retain the same experimental settings as in previous sections (see **Experimental Setup** in Section 5 and **Implementation Details** in Appendix D.1), and vary one adjacency search parameter at a time, including the top-k parameter k, walk length ℓ, number of starting neighbors $r = |R|$, number of permutations P, and the maximum threshold T with the threshold set $L = \{1, 2, \ldots, T\}$. The resulting mean test accuracies are shown in Figures 4 and 5 on 10 representative datasets (heterophilic: *chameleon, cornell, texas, wisconsin*; moderate-degree homophilic: *community, shape*; low-degree homophilic: *citeseer, cora, cycle, grid*).

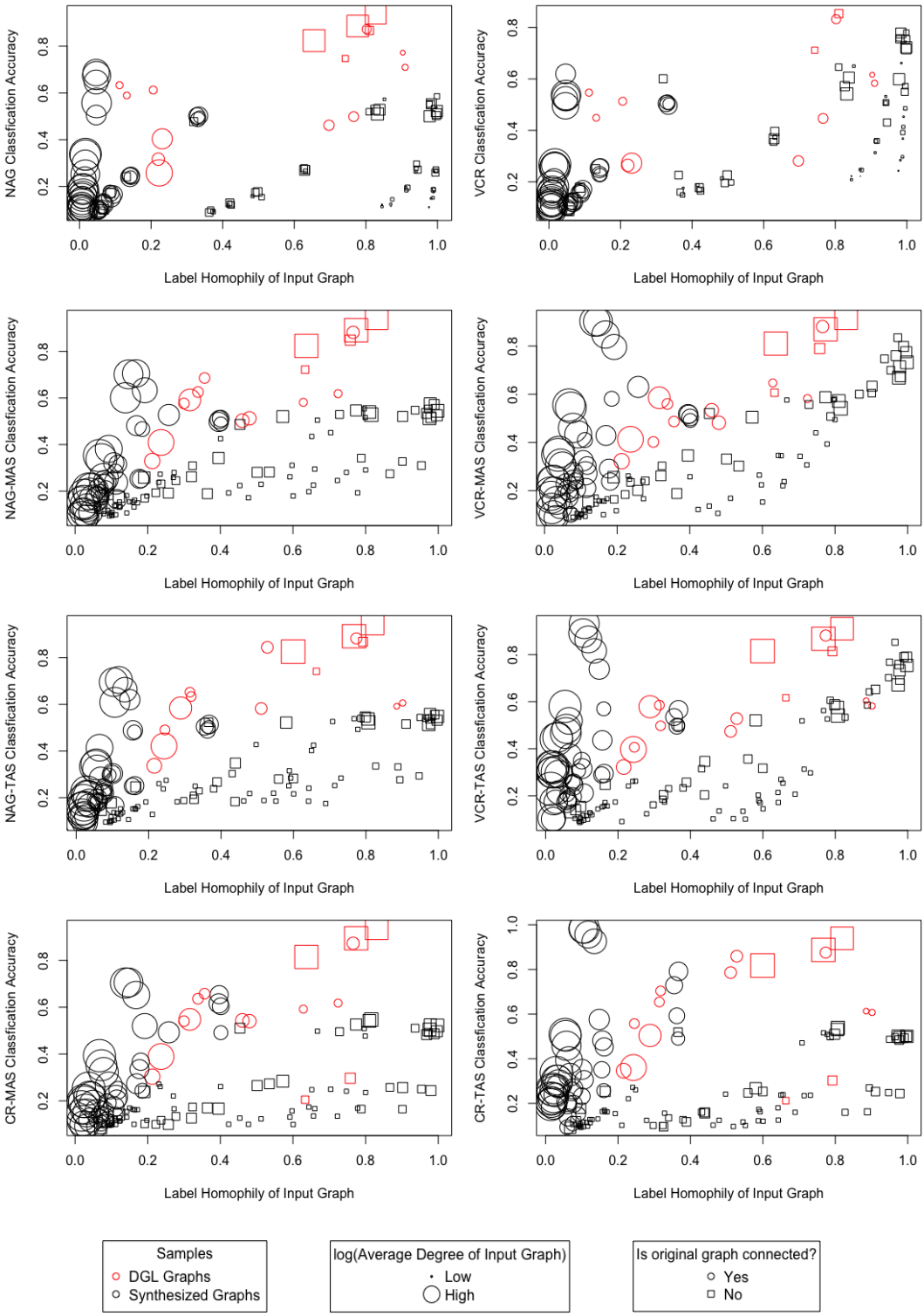

Figure 3: Mean test accuracy vs. structural properties.

CR-MAS (Figure 4).

- **Increasing k (auxiliary graph degree/token length) is generally beneficial.** Increasing k ($3 \to 9$) improves mean accuracy on most datasets, often monotonically, with gains saturating around mid/large k. The trend is especially clear on harder heterophilic/low-degree cases (e.g., *chameleon* rises from $\sim$0.52 to $\sim$0.64; *grid* from $\sim$0.64 to $\sim$0.71), while other high-performing datasets (e.g., *cora*) remain strong and relatively stable (around $\sim$0.80-0.84).
- **Walk length ℓ exhibits diminishing returns (often negative) for MAS.** Longer cascades ($\ell = 5 \to 20$) tend to *reduce* accuracy for many datasets (notably *citeseer*, *grid*, and *shape*), suggesting that MAS benefits most from a *tighter* mesoscopic radius; extending too far likely introduces weaker/less relevant co-activations. This observation is consistent with Theorem 4.8. A few datasets show mild robustness or slight recovery at $\ell = 20$, but the dominant pattern is "smaller ℓ is more stable" for MAS.
- **Starting-neighborhood size r, when large, could degrade performance for MAS.** Varying r ($3 \to 9$) is mostly mild, but some datasets degrade sharply with larger r (e.g., *shape* drops from $\sim$0.63 to $\sim$0.49; *community* from $\sim$0.58 to $\sim$0.50). This indicates MAS is sensitive to noisy initialization when the initial active set is too wide. A safer choice is choosing $r \leq d_{avg}$, where d_{avg} is the average degree in the graph.
- **Number of permutations P is a second-order effect for MAS.** Changing P ($1 \to 7$) yields relatively small shifts on most datasets, with only modest improvements at intermediate values (some datasets peak around $P \approx 3$–5). Overall, MAS appears fairly robust to P once a small amount of resampling is used.

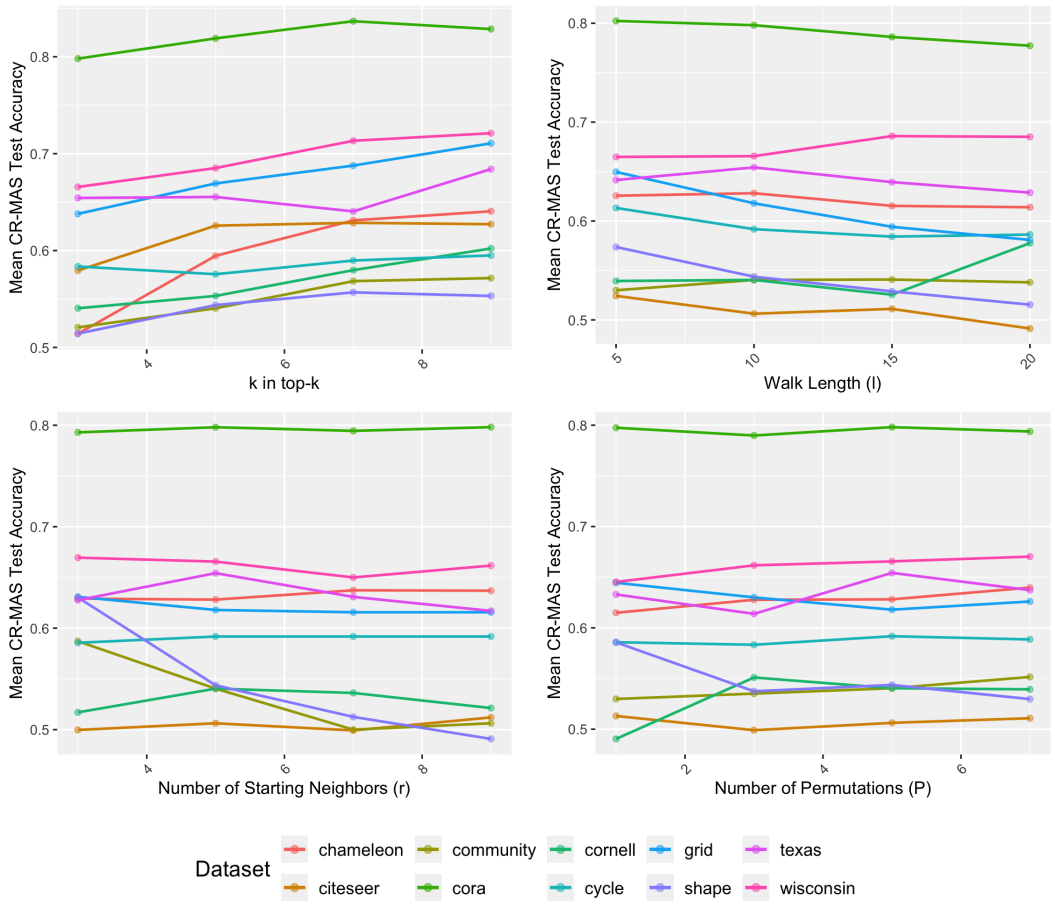

Figure 4: Parameter Analysis for CR-MAS.

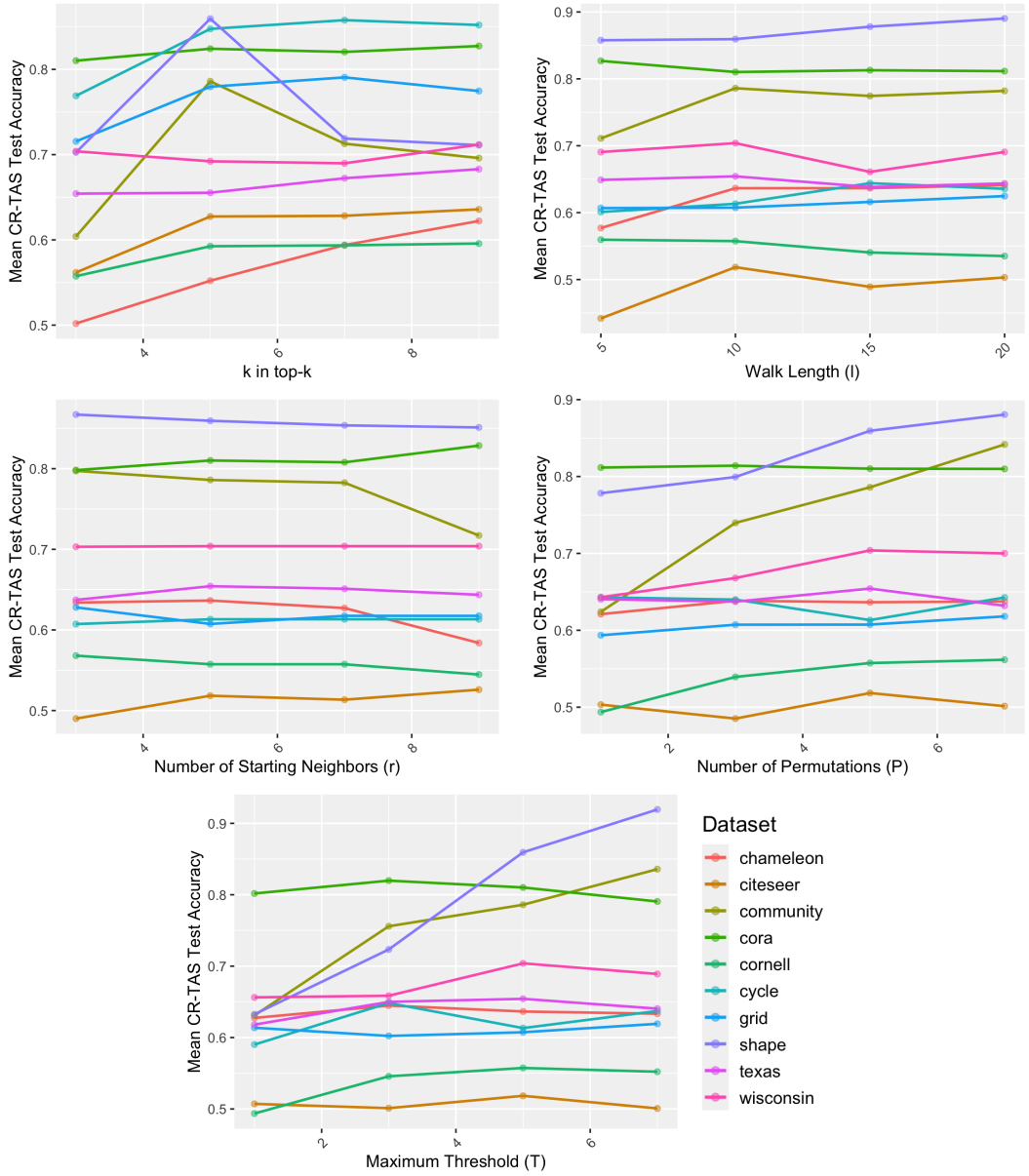

Figure 5: Parameter Analysis for CR-TAS.

CR-TAS (Figure 5).

- **Top-k is non-monotonic for several datasets.** Unlike MAS, TAS shows clear non-monotonicity: some datasets peak at intermediate k (notably *community* and *shape*, which peak at $k \approx 5$ and then drop when k is increased further), while heterophilic *chameleon* improves steadily with larger k ($\sim$0.50 $\rightarrow$ $\sim$0.62). This suggests TAS can over-include less useful neighbors if k becomes too large, depending on the graph type.

- **TAS is more tolerant of longer walk lengths.** Accuracy is typically stable or slightly increasing with ℓ (e.g., *shape* increases from $\sim$0.86 to $\sim$0.89 as ℓ grows; *chameleon* improves mainly from $\ell = 5$ to $\ell = 10$ and then plateaus). Compared to MAS, TAS does *not* show the same widespread degradation as ℓ increases. This observation is consistent with thresholding preventing premature activation of weakly supported nodes.

- **Starting-neighborhood size r is mostly robust, but very large r is generally not preferable.** Most datasets vary only slightly with r, but the largest r setting causes noticeable drops for some (e.g., *chameleon* and *community*), implying that even under thresholding, an overly broad initialization can inject noise. A safer choice is to choose $r \leq d_{avg}$.
- **Permutation count P is more important under TAS than MAS.** Increasing P yields pronounced gains for certain datasets—especially *community* ($\sim$0.62 $\rightarrow$ $\sim$0.84) and *shape* ($\sim$0.78 $\rightarrow$ $\sim$0.88), while others remain comparatively flat. This indicates TAS benefits more from additional resampling/permutation averaging, likely because its thresholded selection introduces more variance that is reduced with repeated runs.
- **Maximum threshold T (i.e., allowing larger reinforcement thresholds) yields greater gains.** Increasing T produces the largest visible ablation effects: *shape* rises dramatically ($\sim$0.63 $\rightarrow$ $\sim$0.92) and *community* increases strongly ($\sim$0.63 $\rightarrow$ $\sim$0.84), whereas several other datasets change modestly and a few show slight saturation/decline at high T (e.g., *cora*). Overall, the figure suggests that enabling stronger multi-neighbor reinforcement (higher thresholds) can be crucial on some graphs, but is not uniformly beneficial across all datasets.

D.6.2 The Use of W^* in Cascaded Graph Learning

Throughout the main body, *cascade models* refer to models that receive the auxiliary graph G^* together with cascade-frequency edge weights W^* as input. We also evaluated the topology-only variant in which only G^* is provided. Tables 12 and 13 show that the effect of W^* is architecture-dependent. For sparse-attention GTs, the topology alone already yields broad gains: VCR improves on 13 of 16 graphs and NAG-MAS on up to 6, with the largest single gain reaching $+33.31\%$ for VCR. For GNN-based architectures (GCN, GraphGPS), the topology-only variant produces fewer improvements (4 and 7 of 16, respectively, with gains up to $+4.30\%$), reflecting that GNN aggregation is more sensitive to degree imbalance and benefits more from weighting. Adding W^* on top of G^* extends the gains for all backbones: at least 10 graphs improve for MAS cascades (up to $+22.79\%$ over the topology-only variant) and at least 11 for TAS cascades (up to $+35.89\%$). Performance declines, when they occur, are small: the only notable drops are for VCR-MAS and VCR-TAS on *cycle* and *grid* (-7.55% to -14.53% relative to VCR with topology-only input) and VCR-MAS on *community* (-6.94%) and *squirrel* (-3.39%); all other declines are at most 1.46%.

Table 12: Number of the 16 DGL graphs whose test accuracy improves when using the auxiliary graph G^*, with or without cascade frequency-based edge weights W^*. Column 3 summarizes Table 13. Column 4 summarizes Table 9. Column 5 summarizes the gains from Table 9 over Table 13.

Baseline	Cascade Mode	Improvement from G^* Input	Improvement from Cascade Model	Improvement of Cascade Model over G^* Input
GCN	MAS	4 (up to **+3.05%**)	8 (up to **+17.11%**)	14 (up to **+16.60%**)
	TAS	3 (up to **+3.23%**)	10 (up to **+20.15%**)	14 (up to **+30.66%**)
GraphGPS	MAS	7 (up to **+4.29%**)	8 (up to **+8.15%**)	10 (up to **+7.02%**)
	TAS	4 (up to **+2.45%**)	9 (up to **+9.52%**)	11 (up to **+12.63%**)
NAG	MAS	6 (up to **+19.18%**)	11 (up to **+19.77%**)	15 (up to **+19.15%**)
	TAS	5 (up to **+18.90%**)	13 (up to **+34.54%**)	14 (up to **+35.89%**)
VCR	MAS	12 (up to **+32.81%**)	13 (up to **+32.74%**)	11 (up to **+22.79%**)
	TAS	13 (up to **+33.31%**)	13 (up to **+32.33%**)	11 (up to **+29.65%**)

These observations can be explained by the fact that the frequency of node activations under MAS and TAS captures the mesoscopic structural information beyond the raw graph topology. Nodes that are repeatedly co-activated with a given seed are typically embedded in the same cohesive cluster or dense region, while nodes in sparser regions or across weak cuts activate less frequently. Using these frequencies as edge weights biases the GNN and GT message-passing mechanisms toward structurally relevant neighbors, amplifying signals from densely connected clusters and downweighting less informative connections. In contrast, simply inputting the auxiliary graph with unit edge weights provides only topological information without distinguishing the relative importance of different edges, which limits the improvement in classification. Consequently, the cascade frequency-weighted graphs enable models to exploit higher-order connectivity and cluster cohesion, resulting in larger and more consistent gains across datasets and architectures.

Table 13: Test accuracy (mean ± std, in %) across DGL datasets where non-baseline methods input auxiliary graphs into the baseline models without frequency-based edge weights. OOM indicates out-of-memory errors. Darker cell colors indicate better performance.

Method	*actor*	*chameleon*	*citeseer*	*community*	*computer*	*cora*	*cornell*	*cycle*
GCN	29.59±0.91	57.30±2.67	75.70±1.41	55.63±1.92	90.30±0.66	87.96±0.83	45.11±6.20	60.73±6.59
GCN-MAS	28.27±0.88	59.12±2.37	70.86±1.75	50.39±2.54	88.25±0.74	83.57±1.05	44.26±5.15	59.22±2.69
GCN-TAS	28.31±0.72	59.99±2.31	73.24±1.28	47.60±2.17	88.93±0.83	86.26±0.89	41.28±6.74	59.27±2.73
GraphGPS	35.19±1.31	60.78±1.35	75.23±1.63	44.00±2.30	OOM	86.65±1.04	61.38±5.49	55.11±10.10
GPS-MAS	35.24±1.32	59.73±1.71	72.63±1.48	47.94±3.11	OOM	85.08±1.48	61.70±6.14	56.26±3.37
GPS-TAS	35.25±1.03	59.55±1.55	73.83±1.42	42.33±1.72	OOM	86.16±1.00	60.43±4.95	57.56±5.66
NAG	31.68±1.26	40.38±1.81	74.72±1.67	46.17±2.49	88.66±0.73	86.75±0.82	58.94±7.11	77.24±9.34
NAG-MAS	33.08±1.41	59.56±1.27	70.82±1.66	49.66±2.15	88.46±0.78	83.69±0.87	56.49±7.26	57.99±2.53
NAG-TAS	32.88±1.14	59.28±1.19	72.67±1.59	42.90±1.52	88.87±0.74	86.05±1.13	44.89±6.36	59.82±2.34
VCR	26.41±1.20	27.28±2.51	71.15±1.36	28.13±4.60	82.45±5.30	85.41±1.33	44.89±6.09	61.64±4.12
VCR-MAS	31.60±1.73	60.09±2.48	63.45±1.77	54.16±2.39	89.32±0.69	80.76±0.92	51.06±6.33	59.25±2.94
VCR-TAS	32.43±1.70	60.59±2.67	66.03±1.72	49.37±1.84	89.56±0.71	82.56±0.96	48.83±5.41	59.20±2.73

Method	*grid*	*photo*	*pubmed*	*shape*	*squirrel*	*texas*	*wiki*	*wisconsin*
GCN	59.34±5.45	93.90±0.74	87.16±0.35	89.46±11.87	39.10±1.43	54.57±7.12	83.89±0.73	51.41±4.81
GCN-MAS	58.19±1.76	91.75±1.09	83.78±0.42	54.69±4.47	40.49±3.42	56.91±8.25	79.72±0.95	54.45±5.72
GCN-TAS	58.19±1.76	92.44±0.83	84.69±0.38	58.91±7.93	42.33±1.83	57.02±6.04	81.34±0.74	51.09±4.03
GraphGPS	57.41±10.00	94.64±0.63	OOM	46.74±14.93	40.04±1.27	69.68±6.58	OOM	72.66±4.95
GPS-MAS	53.48±7.51	94.46±0.63	OOM	48.14±4.45	39.24±1.50	72.45±7.02	OOM	76.95±3.58
GPS-TAS	56.33±5.59	94.06±0.97	OOM	43.63±8.60	38.10±1.27	69.79±7.19	OOM	74.14±5.23
NAG	70.99±1.99	93.94±0.67	87.26±0.52	49.89±6.51	25.80±1.26	63.30±6.43	82.40±0.98	61.25±4.32
NAG-MAS	58.22±1.72	93.45±0.69	88.08±0.41	45.77±4.53	39.11±1.98	62.13±7.06	81.39±0.92	67.73±3.80
NAG-TAS	57.91±1.99	93.68±0.48	88.15±0.40	48.54±4.64	39.72±1.62	61.60±6.60	81.29±0.67	58.28±5.53
VCR	58.32±1.96	88.45±7.85	83.25±0.44	44.66±6.15	22.48±2.00	54.68±7.28	78.46±2.69	51.33±7.46
VCR-MAS	58.58±1.62	92.99±0.80	87.18±0.63	43.11±2.97	42.76±2.58	62.45±8.32	81.94±0.76	67.11±5.97
VCR-TAS	59.24±3.01	93.39±0.71	87.35±0.42	47.29±5.49	43.17±3.30	61.70±7.78	82.86±0.70	65.86±5.83

D.7 Comparison to Existing Rewiring Baselines

The VCR block in Tables 1 and 4 already provides a controlled comparison between Personalized PageRank (PPR) and cascade-based token selection under identical model configurations and training protocols. Replacing PPR token lists with cascade-selected lists yields significant performance improvements, outperforming PPR on $13/16$ DGL benchmarks ($12/12$ graphs in target regimes, up to $+32.74$ points) and $6/7$ additional datasets in Appendix D.9 (up to +39.05%). Additionally, broader PPR and diffusion-propagation methods are already represented in the paper through baselines such as PPRGo and GRAND+.

To further evaluate these mechanisms, we compare GCN cascade models (GCN-MAS and GCN-TAS) against the unrewired GCN baseline as well as strong structural rewiring methods, including GCN+FoSR [Karhadkar et al., 2023] and GCN+SDRF [Topping et al., 2022], across all 16 DGL benchmarks under the experimental protocol described in Section 5.

Table 14: Test accuracy (mean, in %) of GCN cascade models compared with unrewired GCN and structural rewiring baselines GCN+FoSR and GCN+SDRF across all 16 DGL benchmarks. Bold text indicates the best performance per dataset.

Dataset	GCN	GCN+FoSR	GCN+SDRF	GCN-MAS	GCN-TAS
actor	29.59	32.11	31.73	34.28	**34.62**
chameleon	57.30	43.66	44.79	57.67	**58.75**
citeseer	75.70	**76.35**	71.91	74.71	75.46
community	**55.63**	51.09	50.34	51.76	50.11
computer	90.30	83.78	80.11	89.34	**90.87**
cora	**87.96**	86.10	85.84	84.93	87.29
cornell	45.11	50.74	51.54	**60.85**	57.02
cycle	60.73	**68.04**	66.91	59.27	60.39
grid	59.34	70.25	**72.33**	58.30	58.19
photo	93.90	90.21	90.37	93.91	**94.17**
pubmed	87.16	74.86	71.62	87.87	**88.15**
shape	89.46	81.50	79.01	57.69	**89.57**
squirrel	39.10	40.03	38.98	40.01	**42.38**
texas	54.57	55.50	52.25	**66.49**	65.00
wiki	**83.89**	73.91	69.39	81.79	83.54
wisconsin	51.41	57.63	58.78	68.52	**71.56**

As shown by Table 14, cascade-based rewiring significantly outperforms the structural baselines on the vast majority of tasks: GCN-TAS achieves the highest accuracy among all methods on 8 of 16 datasets, while GCN-MAS wins on an additional 2 (*cornell* and *texas*). Combined, cascade models outperform both FoSR and SDRF on 13 out of 16 benchmarks. While FoSR and SDRF are not weak—delivering substantial gains on synthetic bottleneck graphs such as *cycle* (68.04%) and *grid* (72.33%)—they struggle on heterophilic and dense homophilic graphs. In contrast, cascade rewiring yields the largest margins on heterophilic graphs (*wisconsin*, *texas*, *cornell*, *actor*, *chameleon*, *squirrel*) and moderate- to high-degree homophilic benchmarks (*pubmed*, *computer*, *photo*, *wiki*, *shape*).

The few cases where structural baselines win cluster precisely in regimes predicted by our theoretical analysis (Theorem 4.8 and Appendix B.10): bottleneck-dominated graphs where cascade rewiring cannot bridge narrow cuts (*cycle*, *grid*) and low-degree homophilic graphs where same-label pairs lack reinforced support (*citeseer*). These are precisely where FoSR and SDRF are designed to operate. On *community*, all rewiring methods underperform unrewired GCN (55.63%), whereas CR-Graphormer achieves the best overall result with 78.60% test accuracy. Overall, these results demonstrate that cascade operators provide strong, complementary performance gains over classical structural rewiring.

To validate these mechanisms on transformer backbones, we extend the comparison to VCR-Graphormer using the same 20 splits, sparsity budget, and validation protocol, incorporating GDC [Gasteiger et al., 2019] as an additional PPR/heat-kernel sparsification rewiring baseline. We report results on nine of the sixteen benchmarks: all six heterophilic datasets, *pubmed*, and two homophilic datasets (*cora* and *citeseer*) where theory predicts cascade rewiring should be least effective.

Table 15: Test accuracy (mean ± std, in %) of VCR cascade models compared with structural and spectral rewiring baselines VCR+FoSR, VCR+SDRF, and VCR+GDC across DGL datasets. Bold text indicates the best performance per dataset.

Dataset	VCR	VCR+FoSR	VCR+SDRF	VCR+GDC	VCR-MAS	VCR-TAS
actor	26.41 ± 1.20	34.91 ± 1.65	33.03 ± 3.39	28.82 ± 1.40	36.18 ± 1.23	$\mathbf{36.28 \pm 1.42}$
chameleon	27.28 ± 2.51	50.72 ± 4.22	52.69 ± 5.49	40.97 ± 3.64	$\mathbf{60.02 \pm 2.87}$	59.61 ± 1.95
citeseer	71.15 ± 1.36	$\mathbf{74.54 \pm 3.40}$	70.17 ± 1.48	73.35 ± 1.66	71.47 ± 1.64	74.14 ± 1.53
cora	85.41 ± 1.33	$\mathbf{87.19 \pm 3.44}$	85.28 ± 1.22	87.07 ± 1.01	85.08 ± 0.57	85.11 ± 0.62
cornell	44.89 ± 6.09	58.08 ± 3.33	63.10 ± 3.77	62.81 ± 1.47	73.85 ± 5.72	$\mathbf{75.02 \pm 4.68}$
pubmed	83.25 ± 0.44	78.77 ± 3.17	76.49 ± 0.79	85.63 ± 0.91	87.59 ± 0.46	$\mathbf{88.70 \pm 0.91}$
squirrel	22.48 ± 2.00	41.47 ± 3.87	40.50 ± 5.42	39.04 ± 0.95	39.37 ± 1.49	$\mathbf{46.06 \pm 1.04}$
texas	54.68 ± 7.28	67.25 ± 5.91	64.28 ± 5.32	66.01 ± 4.85	80.99 ± 5.63	$\mathbf{82.31 \pm 6.59}$
wisconsin	51.33 ± 7.46	70.89 ± 6.13	69.32 ± 5.45	72.29 ± 2.58	79.93 ± 3.76	$\mathbf{81.00 \pm 2.66}$

Table 15 shows that both operators still help on this backbone. MAS and TAS each improve over the unrewired VCR-Graphormer on eight of the nine datasets; the exception is *cora*, where both fall by up to 0.33%, well within the $\pm 1.33\%$ deviation of the unrewired baseline. Against the best of the three baselines on each dataset, TAS wins on seven and MAS on six. Against GDC alone, eight and seven. Rewiring also reduces variance on the small graphs, from $\pm 7.46\%$ to $\pm 2.66\%$ on *wisconsin* and from $\pm 6.09\%$ to $\pm 4.68\%$ on *cornell*, so the gains are not driven by a handful of favorable splits. While MAS and TAS consistently beat the baselines on target regimes, the baselines themselves are not weak here. FoSR and SDRF improve VCR-Graphormer on eight of nine datasets, by as much as $+25.41\%$ on *chameleon*, and GDC improves it on all nine, by up to $+20.96\%$ on *wisconsin*.

The margins are also statistically separated. Comparing the better cascade operator with the strongest baseline on each dataset (Welch's t over the 20 splits), all seven wins are significant at $\alpha = 0.05$, from $p = 8 \times 10^{-13}$ on *wisconsin* and $p = 5 \times 10^{-13}$ on *pubmed* to $p = 8 \times 10^{-3}$ on *actor*. Of the two losses, only *cora* is significant ($-2.08\%, p = 0.015$); the *citeseer* deficit of 0.40% is not distinguishable from zero ($p = 0.64$).

The two losses are *cora* and *citeseer*, the low-degree homophilic regime where Assumption 4.3 fails and same-label pairs lack the redundant short-path support that Theorem 4.8 requires. That is the regime the other methods are built for, consistent with our observations from Table 14.

Finally, we compare MAS and TAS against a direct common-neighbor baseline—a directed top-k selection based on $(A^2)_{uv} = |N(u) \cap N(v)|$, symmetrized using the same k (rounded average degree)—across both the NAGphormer and VCR-Graphormer architectures.

Table 16: Accuracy improvement ($\Delta\%$) over the unrewired baseline for cascade rewiring versus two-hop common-neighbor rewiring (A^2 proxy) across five heterophilic benchmarks using NAGphormer and VCR-Graphormer backbones.

Dataset	NAG-MAS/TAS	A^2 rewiring + NAG	VCR-MAS/TAS	A^2 rewiring + VCR
actor	$+3.42/+3.42$	$+1.21$	$+9.77/+9.87$	$+3.02$
chameleon	$+18.77/+18.03$	$+8.63$	$+32.74/+32.33$	$+15.09$
squirrel	$+15.09/+16.33$	$+6.78$	$+16.89/+23.58$	$+9.71$
texas	$+17.98/+17.98$	$+7.50$	$+26.31/+27.63$	$+11.55$
wisconsin	$+19.77/+19.77$	$+6.19$	$+28.60/+29.67$	$+14.68$

We observe from Table 16 that the A^2 baseline improves both backbones on heterophilic datasets and recovers, on average, **40.54% of the cascade gain** (between 30.91% and 49.48%). This supports Theorem 4.7, which analyzes this two-hop reinforcement mechanism directly. Cascade rewiring outperforms the A^2 proxy in all ten paired comparisons, with gains that are **2.02–3.27× larger**. Because NAGphormer has the stronger original-graph baseline on all five datasets, this consistent margin suggests that the additional benefit comes from the cascade rewiring mechanism.

D.8 Comparison to Heterophily-Specific Architectures

We run additional experiments on seven state-of-the-art architectures developed to address challenges in heterophily (ACMGCN [Luan et al., 2022], CMGNN [Zheng et al., 2025], GGCN [Yan et al., 2022], GloGNN [Li et al., 2022], H2GCN [Zhu et al., 2020], M2MGNN [Liang et al., 2024], OrderedGNN [Song et al., 2023]) and compare their performance against our methods. Comparing GRAPH CASCADES against this set provides a stronger reference point than GNNs and GTs alone: a competitive result here would indicate that cascade rewiring captures heterophily-relevant structure without architecture-specific modifications.

Table 17: Test accuracy (mean ± std, in %) from heterophily-specific architectures across DGL datasets. "Our Best" indicates the best validation-selected cascade models from Table 9. Top-5 baseline accuracies are shaded, with darker shading indicating better performance.

Model	*actor*	*chameleon*	*citeseer*	*community*	*cora*	*cornell*	*cycle*	*grid*	*shape*	*texas*	*wisconsin*
ACMGCN	33.55±1.32	62.74±1.01	76.13±1.59	42.80±1.82	85.49±0.75	60.85±4.41	57.81±1.43	57.80±2.81	44.34±4.11	78.72±8.64	72.50±2.37
CMGNN	34.94±1.16	60.91±1.12	75.61±1.39	76.06±1.24	84.67±1.59	73.62±6.31	88.40±16.36	80.32±5.97	97.14±0.90	84.68±5.90	83.75±2.84
GGCN	26.76±2.37	34.53±6.29	72.70±1.10	65.89±1.67	79.91±4.01	41.70±9.23	90.59±4.28	65.95±13.40	44.34±4.11	44.26±21.67	36.56±13.60
GloGNN	30.58±1.15	65.51±2.52	75.15±1.74	53.94±2.32	84.05±1.38	62.13±4.61	83.65±2.34	72.04±1.42	20.57±24.07	80.85±6.89	69.69±6.31
H2GCN	34.68±0.98	53.61±1.99	76.25±1.75	66.23±1.74	87.06±0.76	68.51±5.90	48.77±6.31	66.21±7.68	76.80±4.77	85.11±7.52	79.69±2.47
M2MGNN	31.61±1.58	64.70±2.12	70.73±2.49	56.51±2.39	85.26±1.22	51.49±4.85	55.16±6.71	57.80±2.81	34.74±12.04	71.91±10.79	59.06±3.20
OrderedGNN	35.61±1.11	56.07±2.26	76.69±1.13	53.83±2.12	87.47±0.97	57.45±5.42	57.81±1.43	53.59±8.23	36.11±9.90	62.98±3.56	73.44±6.35
Our Best	36.46±1.31	63.65±2.82	75.46±1.46	78.60±3.50	87.29±0.77	75.02±4.68	61.32±3.67	62.25±2.21	89.57±11.79	82.31±6.59	81.02±4.13

Table 17 shows that our best validation-selected methods consistently rank among the top five, with performance gaps of up to 2.8% compared to the best model on most datasets. Specifically, we achieve 1st place on *actor*, *community*, and *cornell*; 2nd on *cora* and *wisconsin*; 3rd on *chameleon* and *texas*; and 5th on *citeseer*. For *shape*, *cycle*, and *grid*, our method ranks 2nd, 4th, and 5th, respectively, with performance gaps of 7.57%, 29.27%, and 18.07% compared to the best-performing method. This is consistent with the observations in Section 5: although *cycle* and *grid* exhibit high homophily, their low average degree violates Assumption 4.3, making our approach less suitable.

D.9 Experiments on Long-Range and "Filtered" Datasets

We extend our experiments to long-range datasets where the prediction task requires capturing global information and structural dependencies between nodes that are separated by large graph distances [Dwivedi et al., 2022, Platonov et al., 2023]. We also repeat the experiments on the filtered version of *chameleon* and *squirrel*, denoted as *chameleon-filtered* and *squirrel-filtered*, as existing literature has suggested that they contain duplicate or near-duplicate nodes that share the same features and neighborhood connectivity, leading to significant data leakage and artificially inflated performance metrics on the original versions of these benchmarks [Platonov et al., 2023]. Table 19 shows the

performance of cascade models in comparison with their direct baselines, while Table 22 presents the performance of our best model versus state-of-the-art GNN and GT benchmarks. Since macro-F1 and AUROC are more robust indicators of performance on these datasets, particularly for long-range tasks where class imbalance can skew standard accuracy and for heterophilic graphs with high label noise [Dwivedi et al., 2022, Platonov et al., 2023], we report these metrics in Tables 20–21 and 23–24. Due to severe class imbalance, AUROC scores could not be computed for *COCO-SP* and *PascalVOC-SP*.

Among the seven selected datasets, *Amazon-ratings*, *Questions*, *Roman-empire*, *chameleon-filtered*, and *squirrel-filtered* are heterophilic graphs, while *COCO-SP* and *PascalVOC-SP* are homophilic graphs with moderate average degree. Experimental results in Tables 18 and 19 indicate that cascades improve performance on most graphs: 6 for both GCN-Cascade and VCR-Cascade models, 5 for NAG-TAS, 4 for NAG-MAS, 3 for GPS-TAS, and 2 for GPS-MAS, with two additional graphs showing unchanged test accuracy for GPS-Cascade models.

Table 18: GRAPH CASCADES performance on 7 newly added datasets. Colors denote the graph type with the largest gain on test accuracy: **red** (heterophilic) and **blue** (moderate- to high-degree homophilic).

Baseline	Cascade Model	# Graphs Improved Out of 5 Target Graphs	Change on Untargeted Dataset *Roman-Empire*
GCN	GCN-MAS	6 (avg. +1.68%, up to **+4.98%**)	-0.45%
	GCN-TAS	6 (avg. +1.60%, up to **+5.17%**)	-2.24%
GraphGPS	GPS-MAS	5 (avg. +0.17%, up to **+1.71%**)	-9.22%
	GPS-TAS	5 (avg. -0.04%, up to **+1.85%**)	-2.10%
NAG	NAG-MAS	3 (avg. +1.10%, up to **+4.69%**)	**+1.00%**
	NAG-TAS	4 (avg. +1.23%, up to **+4.23%**)	**+4.09%**
VCR	VCR-MAS	5 (avg. +2.37%, up to **+10.32%**)	**+39.05%**
	VCR-TAS	5 (avg. +2.83%, up to **+8.88%**)	**+38.95%**

Among the two datasets with duplicate nodes filtered, *chameleon-filtered* benefits from rewiring, as the experiments show no degradation in test accuracy, with the largest gains coming from VCR-MAS and VCR-TAS, at +10.32% and +8.88%, respectively. In addition, although MAS and TAS degrade performance on *Roman-empire* for GNN-based architectures (GCN-MAS: -0.45%; GCN-TAS: -2.24%) and GraphGPS (GPS-MAS: -9.22%; GPS-TAS: -2.10%), GT models benefit substantially from cascade rewiring, as VCR-MAS and VCR-TAS show improvements of +39.05% and +38.95%, while NAG-MAS and NAG-TAS yield +1.00% and +4.09%, respectively. Apart from *Roman-empire*, the performance decline is limited: -4.93% from VCR-MAS on *Amazon-ratings* and -3.12% from GPS-TAS on *squirrel-filtered*, while the degradation from all remaining cascade models across datasets is no more than -1.37%.

Table 19: Test accuracy (mean ± std, in %) across 7 newly added benchmarks. OOM indicates out-of-memory errors. Darker cell colors indicate better performance.

Model	*Amazon-ratings*	*COCO-SP*	*PascalVOC-SP*	*Questions*	*Roman-empire*	*chameleon-filtered*	*squirrel-filtered*
GCN	51.08±1.59	81.60±1.27	67.98±9.07	93.29±0.30	36.81±0.55	40.00±2.42	33.57±1.41
GCN-MAS	53.54±1.59	81.70±1.40	72.96±2.94	93.50±0.36	36.36±1.01	41.79±2.88	34.11±1.40
GCN-TAS	52.74±1.09	81.70±1.40	73.15±1.80	93.50±0.36	34.57±1.22	42.15±1.31	33.90±1.69
GraphGPS	43.58±2.68	81.70±1.40	70.75±0.92	93.49±0.40	66.81±0.28	43.14±3.93	38.49±1.02
GPS-MAS	45.29±3.28	81.70±1.40	70.75±0.92	93.49±0.37	57.59±0.75	43.68±4.23	37.27±0.83
GPS-TAS	45.43±1.34	81.70±1.40	70.75±0.92	93.50±0.36	64.71±1.71	44.13±4.56	35.37±1.41
NAG	47.69±2.63	81.49±1.76	72.28±1.37	93.70±0.36	58.08±1.59	38.83±1.57	33.43±0.60
NAG-MAS	46.32±2.08	82.72±2.00	76.97±5.68	93.56±0.39	59.08±1.72	41.26±2.31	33.21±0.73
NAG-TAS	46.69±0.56	82.39±1.99	76.51±2.39	93.46±0.40	62.17±1.77	41.61±3.06	34.15±1.77
VCR	51.74±0.77	79.85±2.01	73.05±2.69	93.50±0.36	18.65±0.98	33.18±3.55	31.71±1.70
VCR-MAS	46.81±1.40	82.34±1.54	74.39±4.50	93.54±0.37	57.70±1.12	43.50±2.61	36.66±1.39
VCR-TAS	51.35±1.01	82.77±1.03	74.88±6.00	93.52±0.41	57.60±2.19	42.06±2.74	35.44±1.52
CR-MAS	46.30±1.63	81.64±1.43	73.46±1.67	93.51±0.43	55.19±0.97	42.06±3.13	37.56±1.69
CR-TAS	48.07±1.36	81.64±1.70	74.97±1.60	93.76±0.38	59.82±2.09	42.42±3.64	37.16±1.70

The macro-F1 and AUROC scores in Tables 20 and 21 largely follow the trends observed in test accuracy, while also highlighting the critical impact of cascade rewiring on class-imbalanced and long-range datasets. In several cases, gains in macro-F1 and AUROC are more pronounced than those in accuracy. For instance, while GCN-MAS and GCN-TAS show a +0.21% improvement in accuracy on *Questions*, their macro-F1 scores exhibit larger gains of +3.10% and +0.30%, respectively, while AUROC increases by +11.40% and +10.00%. This suggests that the cascade mechanism helps the

Table 20: Mean macro-F1 score across 5 long-range datasets. Darker cell colors indicate better performance.

Model	*Amazon-ratings*	*COCO-SP*	*PascalVOC-SP*	*Questions*	*Roman-empire*
GCN	0.387	0.063	0.138	0.486	0.290
GCN-MAS	0.418	0.070	0.152	0.517	0.257
GCN-TAS	0.402	0.068	0.130	0.489	0.264
GraphGPS	0.292	0.063	0.092	0.492	0.583
GPS-MAS	0.325	0.063	0.092	0.506	0.464
GPS-TAS	0.284	0.063	0.092	0.510	0.557
NAG	0.356	0.074	0.138	0.533	0.444
NAG-MAS	0.335	0.137	0.240	0.528	0.439
NAG-TAS	0.332	0.131	0.254	0.532	0.451
VCR	0.388	0.102	0.196	0.483	0.108
VCR-MAS	0.315	0.124	0.222	0.492	0.442
VCR-TAS	0.385	0.144	0.254	0.491	0.445
CR-MAS	0.336	0.070	0.172	0.505	0.401
CR-TAS	0.318	0.069	0.171	0.543	0.463

Table 21: Mean AUROC across 3 long-range datasets. Darker cell colors indicate better performance.

Model	*Amazon-ratings*	*Questions*	*Roman-empire*
GCN	0.744	0.540	0.855
GCN-MAS	0.771	0.654	0.798
GCN-TAS	0.756	0.640	0.843
GPS	0.689	0.681	0.949
GPS-MAS	0.710	0.661	0.910
GPS-TAS	0.694	0.657	0.941
NAG	0.700	0.702	0.912
NAG-MAS	0.699	0.713	0.910
NAG-TAS	0.705	0.691	0.919
VCR	0.731	0.597	0.722
VCR-MAS	0.711	0.627	0.906
VCR-TAS	0.740	0.642	0.906
CR-MAS	0.709	0.706	0.895
CR-TAS	0.704	0.697	0.913

model more effectively capture minority classes. Similarly, VCR-Cascades improve test accuracy on *Questions* by up to +0.04%, while AUROC increases by at least +3.00%. The two moderate-degree homophilic datasets, *COCO-SP* and *PascalVOC-SP*, show a similar pattern: NAG-Cascades improve test accuracy by up to +1.23% on *COCO-SP* and +4.69% on *PascalVOC-SP*, while macro-F1 improves by at least +5.70% on *COCO-SP* and at least +10.20% on *PascalVOC-SP*. Likewise, VCR-TAS improves test accuracy on *PascalVOC-SP* by +1.83% and macro-F1 by +5.80%.

Table 22: Test accuracy (mean ± std, in %) across 7 newly added benchmarks. "Our Best Validation selected Model" indicates the best validation-selected cascade models from Table 19. OOM indicates out-of-memory errors. Accuracies for the top-5 best-performing baselines are encoded by shading, with darker cells indicating better performance.

		Amazon-ratings	*COCO-SP*	*PascalVOC-SP*	*Questions*	*Roman-empire*	*chameleon-filtered*	*squirrel-filtered*
GNNs	GCN	51.08±1.59	81.60±1.27	67.98±9.07	93.29±0.30	36.81±0.55	40.00±2.42	33.57±1.41
	GraphSAGE	53.70±1.42	77.98±8.54	63.58±15.38	93.43±0.40	66.64±1.27	41.43±5.17	37.63±1.46
	GAT	49.31±1.23	80.74±3.43	62.81±22.57	93.43±0.38	38.31±1.73	40.63±3.87	33.61±1.86
GTs	GT	48.50±4.01	81.70±1.40	70.73±0.92	93.48±0.40	65.89±1.88	40.54±1.67	35.73±1.49
	Gophormer	43.79±1.10	81.70±1.40	70.30±1.45	93.49±0.36	67.35±2.14	44.04±2.75	40.39±2.30
	SAN	50.28±2.33	81.70±1.40	70.76±0.96	93.27±0.47	68.04±1.02	39.91±2.99	34.18±2.90
	GraphGPS	43.58±2.68	81.70±1.40	70.75±0.92	93.49±0.40	66.81±0.28	43.14±3.93	38.49±1.02
	NAG	47.69±2.63	81.49±1.76	72.28±1.37	93.70±0.36	58.08±1.59	38.83±1.57	33.43±0.60
	Exphormer	45.87±2.92	81.62±1.59	72.38±0.83	93.50±0.36	66.57±0.96	43.41±6.85	41.29±2.25
	VCR	51.74±0.77	79.85±2.01	73.05±2.69	93.50±0.36	18.65±0.98	33.18±3.55	31.71±1.70
Heterophily-Specific Architectures	ACMGCN	55.65±1.76	82.52±1.02	72.52±0.92	93.65±0.39	66.89±1.35	37.76±2.09	36.30±1.44
	CMGNN	51.55±1.62	81.65±1.38	61.67±11.50	93.52±0.43	68.75±1.08	43.68±3.62	40.25±2.42
	GGCN	46.07±4.63	81.69±1.38	70.55±0.96	81.97±8.09	44.07±1.29	28.25±8.43	23.16±2.22
	GloGNN	37.56±11.23	0.40±0.90	28.96±37.59	93.46±0.36	27.28±13.91	44.04±3.26	33.79±7.67
	H2GCN	50.32±1.98	81.44±1.42	72.52±2.78	93.49±0.47	67.10±1.30	42.24±4.81	33.75±1.07
	M2MGNN	51.56±1.03	81.67±1.39	67.62±8.04	93.76±0.49	49.85±1.07	40.63±2.51	32.57±1.45
	OrderedGNN	53.62±2.90	82.29±2.41	75.88±8.66	93.61±0.38	67.14±1.16	40.36±3.40	34.61±2.17
Our Validated	**Best Model**	GCN-MAS	VCR-MAS	NAG-MAS	CR-TAS	GPS-TAS	GPS-TAS	CR-TAS
		53.54±1.59	82.34±1.54	76.97±5.68	93.76±0.38	64.71±1.71	44.13±4.56	37.16±1.70

Table 23: Mean macro-F1 score across 5 long-range datasets. "Our Best" indicates the best validation-selected cascade models from Table 19. Darker cell colors indicate better performance.

	Amazon-ratings	*COCO-SP*	*PascalVOC-SP*	*Questions*	*Roman-empire*
GCN	0.387	0.063	0.138	0.486	0.290
GraphSAGE	0.402	0.062	0.090	0.490	0.570
GAT	0.350	0.067	0.119	0.485	0.302
GT	0.343	0.063	0.092	0.485	0.582
Gophormer	0.258	0.063	0.101	0.499	0.596
SAN	0.390	0.063	0.093	0.524	0.593
GraphGPS	0.292	0.063	0.092	0.492	0.583
NAG	0.356	0.074	0.138	0.533	0.444
Exphormer	0.286	0.077	0.146	0.483	0.564
VCR	0.388	0.102	0.196	0.483	0.108
ACMGCN	0.429	0.126	0.177	0.526	0.574
CMGNN	0.376	0.065	0.115	0.494	0.609
GGCN	0.332	0.063	0.093	0.514	0.360
GloGNN	0.143	0.001	0.039	0.485	0.084
H2GCN	0.358	0.078	0.166	0.505	0.593
M2MGNN	0.423	0.063	0.104	0.532	0.396
OrderedGNN	0.445	0.144	0.241	0.516	0.571
Our Best	0.418	0.124	0.254	0.543	0.557

Table 24: Mean AUROC across 3 long-range datasets. "Our Best" indicates the best validation-selected cascade models from Table 19. Darker cell colors indicate better performance.

	Amazon-ratings	*Questions*	*Roman-empire*
GCN	0.744	0.540	0.855
GraphSAGE	0.748	0.654	0.941
GAT	0.730	0.671	0.852
GT	0.716	0.687	0.940
Gophormer	0.678	0.695	0.953
SAN	0.739	0.663	0.951
GPS	0.689	0.681	0.949
NAG	0.700	0.702	0.912
Exphormer	0.700	0.703	0.947
VCR	0.731	0.597	0.722
ACMGCN	0.759	0.642	0.936
CMGNN	0.746	0.656	0.947
GGCN	0.682	0.494	0.849
GloGNN	0.591	0.664	0.698
H2GCN	0.733	0.626	0.943
M2MGNN	0.760	0.684	0.877
OrderedGNN	0.772	0.626	0.941
Our Best	0.771	0.713	0.941

As shown in Table 22, our best validation-selected method achieves the best performance on three datasets (*PascalVOC-SP*, *Questions*, and *chameleon-filtered*) compared to 17 GNN and GT architectures. On *COCO-SP*, *Amazon-ratings*, *squirrel-filtered*, and *Roman-empire*, our best validation selected method ranks 2nd, 4th, 6th, and 11th, respectively, with moderate gaps of 0.18%, 2.11%, 3.73%, and 4.04% from the top-performing methods. Overall, our approach ranks within the top six on all 6 target datasets (heterophilic or moderate- to high-degree homophilic), whereas *Roman-empire* belongs to the bottlenecked/regular graph regime where strong performance is not expected by our theory. Our strong performance is followed by OrderedGNN, which achieves top-five performance on 5 out of 7 datasets but with slightly weaker results, and then ACMGCN and Gophormer, each achieving top-five performance on 4 datasets. The macro-F1 and AUROC scores reported in Tables 23 and 24 again show consistent results with test accuracy in Table 22.

Remark D.1 (Scope on Long-Range Benchmarks)**.** *Long-range graph benchmarks [Dwivedi et al., 2022, Platonov et al., 2023] target a regime where label-relevant signals must propagate across graph bottlenecks — long chains, narrow cuts, or tree-like structures with limited multi-path connectivity. This is complementary to the regime targeted by* GRAPH CASCADES*: as established in Theorem 4.8, threshold activation certifies κ edge-disjoint short paths from the seed, and such redundancy is especially scarce in bottlenecked structures. We therefore expect cascade rewiring to be most effective on long-range datasets whose informative dependencies are carried by cohesive mesoscopic regions (e.g., PascalVOC-SP and COCO-SP), and less effective on chain-like or grid-like structures (e.g., Roman-empire). The results above are consistent with this prediction: cascades improve macro-F1 substantially on PascalVOC-SP and COCO-SP, remain competitive on Amazon-ratings and Questions, and show inconsistent gains on Roman-empire, where the method's structural assumptions are not satisfied. We report these results not as uniform improvements but as a calibration of the regimes where the mechanism applies.*

D.10 Broad Backbone Evaluations

To broaden the within-backbone evaluation, we ran six additional compatible backbones (GAT, GT, Gophormer, ACMGCN, H2GCN, and GloGNN) on target regimes (heterophilic and moderate- to high-degree homophilic) under the same protocol for cascade models, bringing the total from four to ten (in addition to GCN, GraphGPS, NAGphormer, and VCR-Graphormer in Section 5). We selected models that can use the cascade weights W^* directly without architectural changes. Models that cannot consume edge weights are still within scope: Appendix D.6.2 shows that the unweighted graph G^* can also improve sparse-attention GTs, and CR-GRAPHORMER is designed for this setting. However, these models do not permit the same controlled weighted comparison.

Table 25: Number of selected baselines (out of 10: GCN, GraphGPS, NAGphormer, VCR-Graphormer, GAT, GT, Gophormer, ACMGCN, H2GCN, and GloGNN) where GRAPH CASCADES improves test accuracy per dataset.

	MAS			TAS		
Dataset	**# Models Improved**	**Average Change**	**Max Change**	**# Models Improved**	**Average Change**	**Max Change**
actor	9	+2.44%	+9.77%	9	+2.53%	+9.87%
cornell	9	+7.62%	+28.96%	8	+7.95%	+30.13%
wisconsin	7	+7.97%	+28.60%	8	+8.83%	+29.67%
chameleon-filtered	8	+1.92%	+12.89%	7	+1.62%	+9.30%
COCO-SP	7	+0.27%	+0.94%	8	+0.37%	+1.59%
squirrel-filtered	6	+0.87%	+3.73%	8	+0.55%	+3.34%
Questions	6	+0.64%	+3.71%	6	+0.78%	+3.69%
shape	5	-2.34%	+25.14%	6	+9.96%	+39.34%
texas	4	+3.01%	+26.31%	5	+3.98%	+27.63%

Table 25 shows that, across 10 backbones, both cascade operators—MAS and TAS—demonstrate robust and consistent performance gains on the majority of evaluated datasets, notably on *actor, cornell, wisconsin, chameleon-filtered, COCO-SP,* and *squirrel-filtered*. The gains are most pronounced on *cornell* (up to $+7.95\%$ on average and $+30.13\%$ in individual gain), *wisconsin* (up to $+8.83\%$ on average and $+29.67\%$ in individual gain), *shape* (up to $+9.96\%$ on average and $+39.34\%$ in individual gain), and *texas* (up to $+3.98\%$ on average and $+27.63\%$ in individual gain). Although

TAS outperforms MAS in most of the evaluations, the magnitude of performance gains remains consistent per dataset across both contagion operators.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims made in the abstract and introduction are justified with theoretical results and empirical experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A paragraph towards the end of main text discusses the limitations of our approach.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All the necessary assumptions and proofs are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper discloses the training, validations and test splits together with hyperparameters used with a detailed ablation study.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a GitHub URL containing the source code. All datasets are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the necessary details together with the code(url) is provided in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report standard deviations and conduct regression analysis (Table 2) to show that our rewiring indeed increases homophily in networks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the appendix, we provide sufficient information about compute resources and hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper studies a rewiring architecture in Graph Transformers which can be used as a black box in many applications. To the best of our knowledge we are not aware of direct societal harms of our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: To the best of our knowledge, the paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper or library that produced the datasets or code.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code using a GitHub URL for replication purposes.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing or research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are used only for writing and grammar checking purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.